

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Trabajo Fin de Grado

ING. ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

PROGRAMACIÓN DE ROBOT KUKA PARA PALETIZADO

AUTOR: Javier de Pinto Hernández

TUTOR: Alberto Jardon Huete

Agradecimientos

Me gustaría aprovechar estas líneas para agradecer a todas las personas que me han apoyado y que han confiado en mí durante esta etapa tan importante de mi vida.

Debo dar gracias a mi familia; a mi madre por confiar en mis posibilidades de graduarme como ingeniero, a mi padre por apoyarme y por saber cómo ayudarme, y a mis hermanas por la confianza que han tenido en mí y por su ayuda.

Me gustaría agradecer también a todos mis amigos, especialmente a los cuatro que han formado parte de mi vida académica y con los que he compartido este viaje. Siempre han estado dispuestos a ayudarme y me llevo cuatro amigos para siempre, gracias Jesús, Samuel, Iván y Bea.

Quiero dedicar unas palabras a mi tutor Alberto Jardon Huete por ofrecerse a supervisar mi trabajo y por confiar en mí para alcanzar tal objetivo. Gracias por la ayuda proporcionada y por los problemas que me ha solucionado.

No quiero olvidarme de agradecer la ayuda brindada por Ismael Huete Picazo. Gracias a él estoy realizando este proyecto y sin su colaboración no hubiera sido posible.

En resumen dar gracias a todas las personas por las que ha sido posible realizar este trabajo.

Resumen

El presente Trabajo Fin de Grado pretende llevar a cabo un sistema muy extendido dentro del mundo de la logística como son los sistemas de paletizado.

Hoy en día los sistemas de paletizado están presentes en todos los ámbitos del sector industrial. Son muchas las empresas que hacen uso de alguno de estos sistemas dentro de su cadena productiva. Los sistemas de paletizado han supuesto un gran avance para la industria.

Para realizar el proyecto se ha seleccionado un sistema robotizado con un brazo robótico de cuatro ejes de la marca KUKA KR 180 3200 PA y la unidad de control del robot KR C4 (KUKA System) que se encarga de “dar vida” al brazo robotizado. La unidad de control de robot estará integrada en un sistema automatizado de transporte gobernado por un autómata industrial PLC de la marca OMRON, concretamente el modelo CJ2M.

Abstract

The following Final Grade Project tries to develop a very widespread system into the logistic world, the palletizing systems.

Nowdays the palletizing systems are inside all the areas of the industrial sector. There are a lot of companies that use one of this systems somewhere in their productive chain. The palletizing systems represent a great improvement for the industry.

It has been selected a robotic system with a robotic arm of four axis of the brand KUKA to build this project, specifically the model KR 180 3200 PA. Also the control unit KRC4 (KUKA System) handle the robotic arm. This control unit is integrated in an automatic transport system lead by an industrial PLC of the brand OMRON, specifically the model CJ2M.

Contenido

Agradecimientos	2
Resumen.....	3
Abstract	4
Índice de ilustraciones.....	7
Índice de tablas	9
1. Motivación	10
2. Objetivos	11
2.1. Estado actual de la instalación	11
2.2. Objetivos por parte del cliente.....	12
2.3. Producto a paletizar	13
3. Estado de la técnica.....	16
3.1. ¿Qué es la paletización?	16
3.2. Ventajas de la paletización.....	17
3.3. Sistemas de paletizado automático	17
3.4. Sistema de transporte previo y posterior al paletizado	19
4. Anteproyecto	20
4.1. Solución inicial.....	20
4.1.1. Transporte máquina 1	21
4.1.2. Transporte máquina 2	22
4.1.3. Alimentador de palets	23
4.1.4. Mesa de paletizado y almacenaje	24
4.1.5. Herramienta cogida.....	26
4.1.6. Control del sistema	28
5. Robot industrial KUKA.....	29
5.1. Elementos de un robot industrial.....	29
5.1.1. Manipulador KR 180 R3200 PA	30
5.1.2. Unidad de control del robot KR-C4	34
5.1.3. Unidad manual de programación KUKA smartPAD.....	38
5.2. Sistema de coordenadas	40
5.2.1. Sistema de coordenadas WORLD	40
5.2.2. Sistema de coordenadas ROBROOT	41
5.2.3. Sistema de coordenadas BASE	41
5.2.4. Sistema de coordenadas TOOL	42
5.2.5. Movimientos manuales.....	42
5.3. Movimientos del robot.....	44

5.3.1.	Posicionamiento aproximado.....	44
5.3.2.	Tipo movimiento PTP	46
5.3.3.	Tipo movimiento LIN	47
5.3.4.	Tipo de movimiento CIRC.....	48
5.3.5.	Tipo de movimiento Spline	49
5.4.	Programación lógica.....	50
5.4.1.	Grupos de usuario	50
5.4.2.	Estructura de un programa	50
5.4.3.	Tipos de datos	51
5.4.4.	Declaración de variables	51
5.4.5.	Operadores.....	52
5.4.6.	Control de ejecución del programa.....	54
6.	Descripción del software del robot.....	59
6.1.	Configuración Automático Externo	59
6.1.1.	Entradas.....	60
6.1.2.	Salidas.....	60
6.1.3.	Secuencia de inicio	61
6.2.	Configuración de los mosaicos.....	62
6.2.1.	Productos de la línea 1	62
6.2.2.	Productos de la línea 2	65
6.3.	Lista de señales con el PLC	67
6.3.1.	Entradas.....	67
6.3.2.	Salidas.....	69
6.4.	Estructura del programa	70
6.4.1.	Programa coger línea 1 y coger línea 2	71
6.4.2.	Programa coger europalet y palet doble	72
6.4.3.	Programa dejar paquete	73
6.4.4.	Programa buscar palet	74
6.4.5.	Programas de interrupción	74
7.	Presupuesto	75
8.	Conclusiones	77
9.	Bibliografía y referencias.....	78
10.	ANEXOS	79

Índice de ilustraciones

Ilustración 1 Estado actual de la instalación	11
Ilustración 2 Solución de transporte	12
Ilustración 3 Evolución paletización	16
Ilustración 4 Almacenamiento	17
Ilustración 5 Transporte	17
Ilustración 6 Robot Antropomórfico	18
Ilustración 7 Pinza robot	18
Ilustración 8 Solución inicial	20
Ilustración 9 Transporte línea 1	21
Ilustración 10 Transporte línea 2	22
Ilustración 11 Alimentador palets	23
Ilustración 12 Mesa paletizado	24
Ilustración 13 Línea almacenaje	25
Ilustración 14 Brazos garra	26
Ilustración 15 Palpador garra	26
Ilustración 16 Palas de la pinza	27
Ilustración 17 Autómata PLC	28
Ilustración 18 Estación remota NX	28
Ilustración 19 Sistema robot	29
Ilustración 20 Brazo robotizado	30
Ilustración 21 Direcciones ejes	32
Ilustración 22 Zona de trabajo	33
Ilustración 23 Unidad de control	34
Ilustración 24 Elementos KRC4	34
Ilustración 25 Frontal del KRC4	36
Ilustración 26 Vista delantera smartPAD	38
Ilustración 27 Vista trasera smartPAD	38
Ilustración 28 Configuración Space Mouse	39
Ilustración 29 Sistemas de coordenadas	40
Ilustración 30 Sistema WORLD	40
Ilustración 31 Movimientos manuales	42
Ilustración 32 Movimientos Space Mouse	43
Ilustración 33 Desplazamiento ejes	43
Ilustración 34 Aproximación PTP	44
Ilustración 35 Aproximación LIN y CIRC	45
Ilustración 36 Movimiento PTP	46
Ilustración 37 Formulario PTP	46
Ilustración 38 Movimiento LIN	47
Ilustración 39 Formulario LIN	47
Ilustración 40 Movimiento CIRC	48
Ilustración 41 Formulario CIRC	48
Ilustración 42 Ventajas Spline	49
Ilustración 43 Desventajas LIN/CIRC	50
Ilustración 44 Estructura programa .SRC	51
Ilustración 45 configuración automático externo	59

Ilustración 46 Secuencia inicio Automático Externo	61
Ilustración 47 Diagrama flujo principal()	70
Ilustración 48 Diagrama flujo coger_paquete()	71
Ilustración 49 Diagrama flujo coger_palet().....	72
Ilustración 50 Diagrama flujo dejar_paquete	73
Ilustración 51 Diagrama flujo buscar_palet()	74

Índice de tablas

Tabla 1 Productos línea 1	14
Tabla 2 Productos línea2	15
Tabla 3 Datos de los ejes	32
Tabla 4 Frontal del KRC4	37
Tabla 5 Vista trasera smartPAD.....	38
Tabla 6 Vista delantera smartPAD	38
Tabla 7 Formulario PTP	46
Tabla 8 Formulario LIN	47
Tabla 9 Formulario CIRC.....	48
Tabla 10 Operadores aritméticos.....	52
Tabla 11 Resultados operadores aritméticos.....	52
Tabla 12 Combinación operadores geométricos	52
Tabla 13 combinación operadores geométricos.....	52
Tabla 14 Operadores de comparación	53
Tabla 15 Operadores lógicos.....	53
Tabla 16 Combinaciones operadores lógicos.....	53
Tabla 17 Desplazamientos Trigger	57
Tabla 18 Entradas Automático Externo	60
Tabla 19 Salidas Automático Externo.....	60
Tabla 20 Listado Entradas PLC.....	68
Tabla 21 Listado salidas PLC	69
Tabla 22 Costes compras.....	75
Tabla 23 Costes material	75
Tabla 24 Costes mano de obra mecánica.....	75
Tabla 25 Costes mano de obra eléctrica	76
Tabla 26 Costes Ingeniería planos eléctricos, mecánicos y automatización.....	76
Tabla 27 Costes finales	76

1. Motivación

Antes de entrar en la Universidad Carlos III de Madrid y cursar Ingeniería Electrónica Industrial y Automática siempre me había fascinado el mundo de la robótica. Me quedaba embobado viendo en la tele al robot ASIMO de Honda o viendo las plantas industriales de fabricación de vehículos. Esto junto a mi afición por montar y desmontar aparatos para comprender su funcionamiento me llevaron a elegir esta carrera universitaria.

Una vez dentro de la universidad me di cuenta de las posibilidades infinitas que brinda el mundo de la robótica. Poco a poco la robótica va formando más parte de nuestras vidas y no solamente está limitado a un uso por parte de la industria del automóvil. Vemos robots humanoides que pueden interactuar con las personas de forma amigable. Sobre todo en el sector industrial es dónde ha supuesto un gran avance. La mayoría de las empresas hacen uso de algún tipo de robot en la cadena de producción.

Tuve la suerte de realizar prácticas en empresa en la sociedad ASA SL, una empresa dedicada a dar soluciones automáticas principalmente al sector industrial. En mi estancia allí, un cliente nos planteó un problema en el paletizaje de su planta de rollos de papel.

2. Objetivos

El este apartado vamos a definir los objetivos a los que queremos llegar con este proyecto. Todo debe quedar bien definido para conseguir un proyecto que satisfaga las necesidades del cliente.

El objetivo del este proyecto es automatizar el sistema de paletizado usado por el cliente mediante el uso del brazo robótico **KUKA KR 180 3200 PA** en comunicación con el autómatas **CJ2M**.

2.1. Estado actual de la instalación

Las instalaciones del cliente lejos de ser unas instalaciones modernas, cosa habitual en los pequeños fabricantes, paletizaba sus productos empleando mano de obra. Los empleados eran los encargados de depositar los fardos de papel en los palets para su almacenaje.

La instalación actualmente cuenta con dos máquinas empaquetadoras de rollos de papel que surten fardos de papel para ser paletizados con un mosaico específico. El transporte desde la salida de las máquinas hasta la zona de paletizado se hacía con mesas de rodillos locos que por efecto de la gravedad transportaban los paquetes.

Actualmente en cada línea trabajan 4 personas en dos turnos. A partir de ese momento solo trabajará un operario encargado de manejar el sistema desde la pantalla táctil, mediante un toro de carga, proveer de palets al sistema y sacar los palets paletizados para su almacenaje.

La máquina 1 se utiliza para productos de uso industrial, que son más pequeños y duros. La máquina 2 se utiliza para productos domésticos que son más blandos y pueden deformarse. Las máquinas no pueden trabajar simultáneamente, es decir, vamos a paletizar un tipo de producto cada vez.

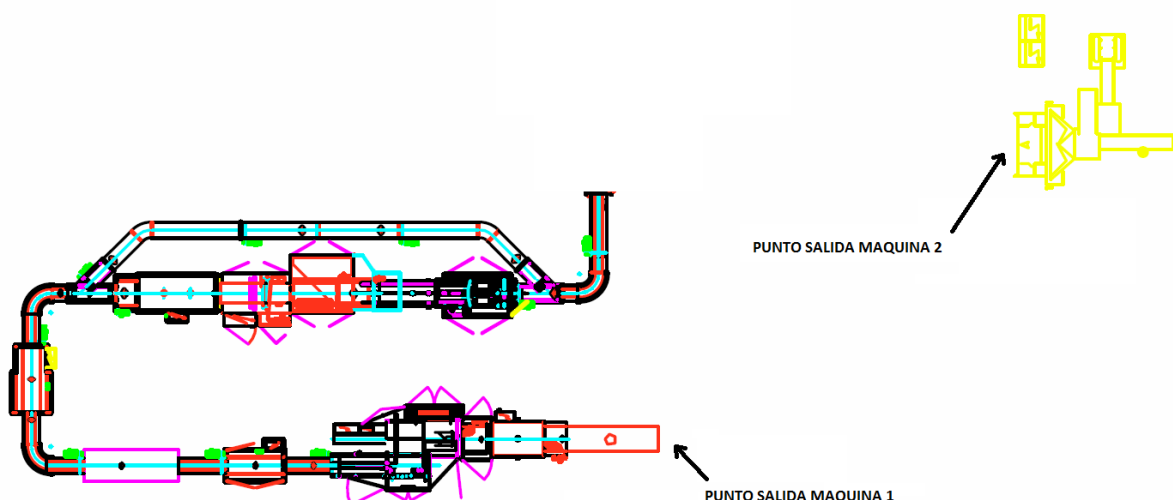


Ilustración 1 Estado actual de la instalación

2.2. Objetivos por parte del cliente

El cliente tenía una idea clara de lo que necesitaba. Había que diseñar un sistema de paletizado con dos líneas de cogida de producto, una línea de cogida de palet y una línea de salida.

Para el transporte de los productos desde la salida de las máquinas hasta los puntos de cogida del robot el cliente nos definió una oferta que garantizaba sus requisitos. La salida de la máquina 1 estaba muy próxima al punto de cogida por lo que no había ningún problema, pero la salida de la máquina 2 queda bastante distante del punto de cogida. Por este motivo y porque necesitan hacer pasar un toro de carga por el espacio de transporte se ha optado por llevar los paquetes mediante cintas de banda y rodillos transportadores elevados.

Debíamos gestionar la producción de las dos líneas con un robot definido por ellos, un KUKA paletizador, concretamente el modelo **KR 180 3200 PA**.

El sistema tiene que ser muy versátil para adaptarse a la variedad de los productos a paletizar y al hecho de que pueden almacenarse en dos medidas de palet distintas.

Los productos de la máquina 1 sólo llevan la etiqueta de identificación por un lado del producto, y un requisito del cliente era que en el paletizado, esa cara siempre estuviera en la parte exterior del mosaico.

En el siguiente esquema mostramos la solución aportada por el cliente para el transporte de los productos de las máquinas.

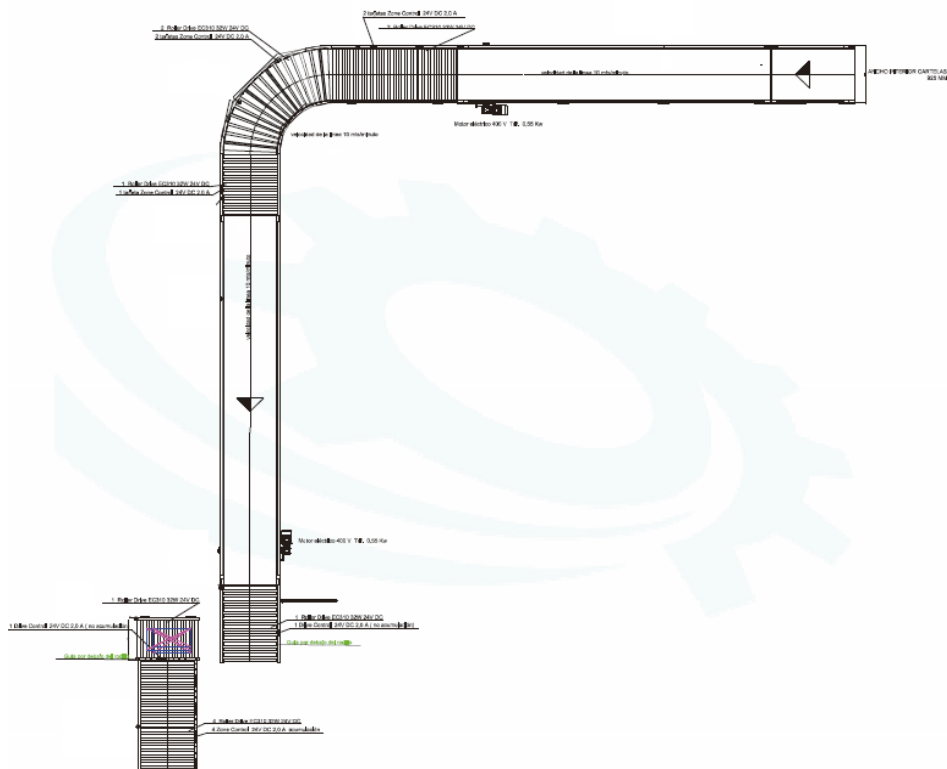


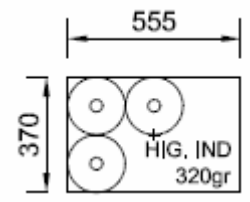
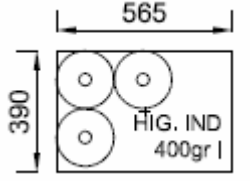
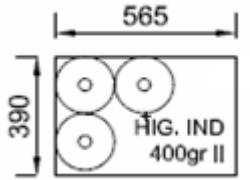
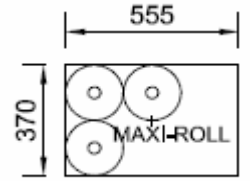
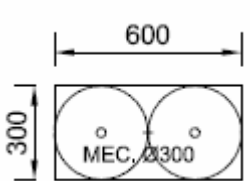
Ilustración 2 Solución de transporte

2.3. Producto a paletizar

Uno de los puntos más importantes dentro del paletizado es conocer el producto que vamos a tratar para poder darle la mejor opción de manipulación. Normalmente en la robótica se suelen utilizar elementos neumáticos o de succión, evitando los motores. Esto es así porque los motores tienen un gran peso y una herramienta de un brazo robótico será mejor cuanto menos peso tenga y cargas más conocidas.

Uno de los grandes retos de este proyecto era la gran variedad en los formatos de los productos que teníamos que paletizar. Desde la máquina 1 saldrían 10 tipos de fardos de rollos de papel para uso industrial, y desde la máquina 2 saldría 5 tipos de fardos de rollos de papel para uso doméstico. Esto hace un total de 15 tipos de productos diferentes en largo, ancho y alto.

Productos de la línea 1:

Nombre producto	Medidas (LARGOxANCHOxALTO)	Esquema
Papel Higiénico Industrial 320gr 18 rollos	(555x370x273)	
Papel Higiénico Industrial 400gr 12 rollos	(565x390x182)	
Papel Higiénico Industrial 400gr 18 rollos	(565x390x273)	
Maxi-Roll 600gr	(555x370x210)	
Papel mecánico Ø300	(600x300x230)	

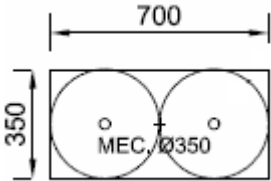
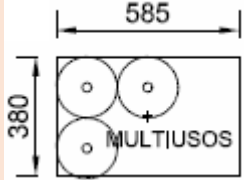
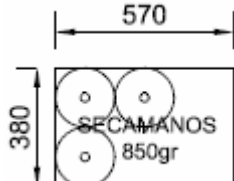

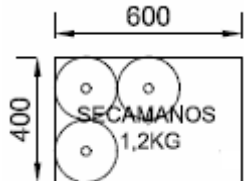
Papel mecánico Ø350	(700x350x230)	
Multiusos individual 6 rollos	(585x380x210)	
Secamanos 850gr	(570x380x195)	
Secamanos 1kg	(600x400x195)	
Secamanos 1.2kg	(600x400x210)	

Tabla 1 Productos línea 1

Productos de la línea 2:

Nombre producto	Medidas (LARGOxANCHOxALTO)	Esquema
Papel Higiénico 15m	(730x580x182)	
Papel Higiénico 16m	(780x585x182)	
Papel Higiénico 18m	(780x858x182)	
Papel Higiénico 22m	(800x600x182)	
Papel Higiénico 35m	(860x660x182)	

Tabla 2 Productos línea2

3. Estado de la técnica

En la realización de un proyecto, una ayuda importante es conocer todo lo relacionado con dicho proyecto, por lo que vamos a explicar los puntos importantes de la paletización.

3.1. ¿Qué es la paletización?

La paletización forma parte de los sistemas de manipulación que más se han desarrollado en las dos últimas décadas. Consiste en agrupar sobre una superficie (palet o estriba) una cierta cantidad de productos individualmente poco manejables, pesados y/o voluminosos, o bien objetos fáciles de desplazar pero numerosos, cuya manipulación y transporte requerirían de mucho tiempo y trabajo. La finalidad es llevar esta mercancía al punto deseado, con el mínimo esfuerzo posible y en una sola operación. Su nombre se debe a la plataforma utilizada como soporte.

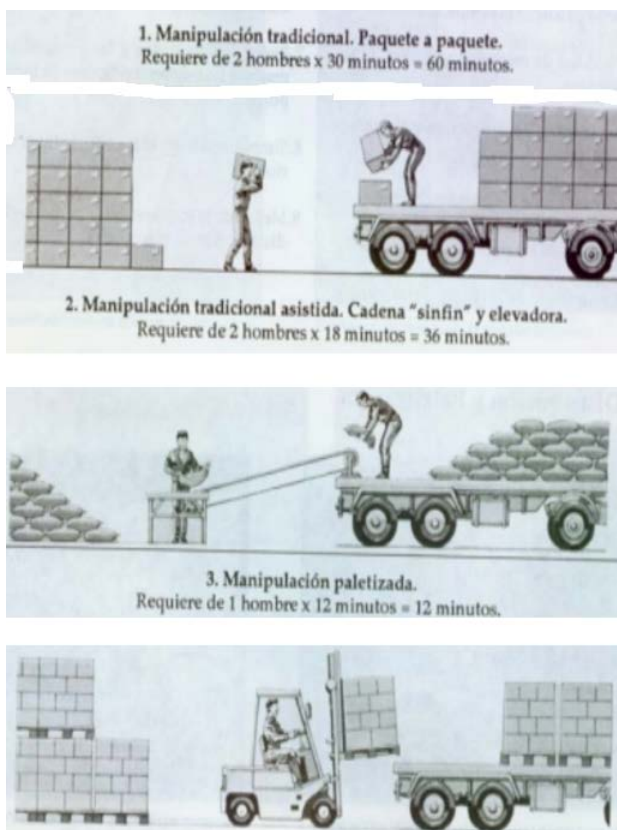


Ilustración 3 Evolución paletización

En sus comienzos, los paquetes se colocaban manualmente sobre las paletas. Los trabajos de paletización eran por tanto manuales y se estudiaban de forma que los operarios realizaran un mínimo de movimientos y consiguieran buenos rendimientos. Pero este trabajo era agobiante y rápidamente estos puestos de trabajo han ido transformándose, volviéndose menos pesados y mejorando las condiciones. Los ingenieros han ido perfeccionando estos puestos manuales, mejorándolos cada vez más, hasta que en la actualidad han hecho su aparición en el mercado los inicialmente semiautomáticos (en los cuales el operario debía efectuar un trabajo de ordenamiento o de control mecánico), y después totalmente automáticos, esto es, que pueden trabajar sin la presencia de un operario.

La paletización se ha desarrollado en todos los sectores de la actividad industrial. La diversidad de productos, las cantidades producidas y los modos de producción han dado lugar a una multitud de tipos de paletizadores.

3.2. Ventajas de la paletización



Ilustración 4 Almacenamiento

Disminución de los tiempos de preparación y carga de vehículos.

Menores costes de carga y descarga.

Aumento de la productividad.

Menor manipulación de los productos.

Posibilidad de prácticas de reabastecimiento continuo.

Optimización del espacio disponible y facilidad de rotación de lo que se almacena.

Disminución de averías por la manipulación de los productos.

Mayor seguridad para el personal involucrado en el manejo de mercancías.

Mejora de la imagen de los productos en el punto de venta y en el almacén.



Ilustración 5 Transporte

Uso más eficiente de la flota de transporte

3.3. Sistemas de paletizado automático

Existen diversos sistemas de paletizado automático mediante robots o células de paletizado automático.

Según las unidades producidas y número de referencias que se desea paletizar, se puede elegir entre diferentes tipos de sistemas. Por otro lado, es determinante el sistema de toma de los productos, ya que una toma del producto individual o de varios productos, o de la capa completa, va a determinar la cadencia final del sistema de paletizado automático.

De esta forma debemos tener en cuenta las siguientes consideraciones:

- Los **robots articulados o antropomórficos** están indicados para manipular un número de referencias más reducido, en un menor espacio, en comparación con el tipo pórtico. La configuración de los palets se desarrolla alrededor del robot, no siendo lineal como ocurre en el pórtico. La cadencia alcanzada con un robot antropomórfico suele ser mayor, en cuanto que los recorridos son menores y están más optimizados.



Ilustración 6 Robot Antropomórfico

- Un **robot pórtico** ofrece un movimiento lineal en 3 ejes (x,y,z definidos como largo, ancho y alto). Esta configuración permite disponer de un mayor número de estaciones de paletizado en línea, por lo que permite paletizar más referencias para un mismo espacio y contemporáneamente con un solo robot. El número de ciclos varía en función de los recorridos a realizar, dimensiones producto y mosaico, entre otros. De esta forma, el pórtico es más flexible en cuanto al espacio útil y un solo cabezal puede recorrer distancias de 30 x 5 x 2,5 metros.
- La concepción de la **pinza del robot** depende de varios factores; mosaicos a formar, número de cajas en toma múltiple, tipo de cartón de las cajas, fragilidad del producto, distribución de su peso en el interior, calidad del precintado etc.



Ilustración 7 Pinza robot

De manera general, podemos clasificar los sistemas de toma en:

- *Pinza de palas*: Formada por la combinación de palas móviles y fijas.
- *Ventosas de vacío*: Cada ventosa incorpora un sensor de apertura y cierre que permite realizar vacío únicamente a las ventosas que se encuentran sobre las cajas.
- *Pinza de garras*: Usualmente utilizadas para la toma de palets.
- *Pinzas mixtas*: Basadas en la combinación entre los sistemas anteriores, según la aplicación.

3.4. Sistema de transporte previo y posterior al paletizado

Automatizar un proceso industrial puede tener como objeto solucionar un aumento de producción, un cambio de producto, implantación de nuevas líneas, buscar una mayor gestión y organización, hasta la conexión de diferentes zonas, creación de almacenes automáticos, organización de las mercancías según destinos.

- **Sistemas de transporte automático de productos ligeros**: Sistemas formados por transportadores de rodillos, bandas, que se integran con sistemas de clasificación como empujadores, sistemas de ruedas direccionales pop-up y bandas verticales o brazos desviadores, elevadores, para conformar el layout de la instalación.

Las instalaciones automáticas se diseñan teniendo en cuenta diferentes parámetros, como las dimensiones del producto, la producción a tratar, procesos, espacio disponible, cuellos de botella, sistemas de entrada y salida del producto, ubicación del código de barras y su lectura automática, ergonomía del operario, etc.

- **Sistemas de transporte automático para palets**: Sistemas automáticos para el transporte de cargas hasta 2.500 Kg., como palets, contenedores, bobinas de papel, o elementos transportados en este tipo de sistemas.

Se forman integrando transportadores de rodillos para palets, de cadenas, transferidores, mesas de giro, carros automotores “shuttle cars”, sistemas de gálibo para palets y pilas de palets. Los sistemas de gálibo para palets, así como el control de tacos y básculas, son imprescindibles en la implementación un almacén automático o situaciones similares, ya que posteriormente serán transportadas en los llamados transelevadores automáticos.

- **Almacén automático, semiautomático o manual**, tanto para cajas, contenedores o palets.

Almacenaje automático vertical con transelevadores de palets y tipo mini-load, o bien *Almacenaje automático horizontal*, haciendo uso de diferentes soluciones como pueden ser la automatización por medio de transportadores convencionales, carros automotores, estanterías dinámicas u otros medios.

4. Anteproyecto

Una vez conocidos los objetivos marcados por el cliente y todos los puntos necesarios para la realización del proyecto, comenzaremos a realizar un anteproyecto para dar una solución inicial 100% funcional, aunque no definitiva.

El principal propósito es solucionar cada uno de los hitos marcados por el cliente y dar una solución global que satisfaga completamente. Este paso es el más importante, permitirá dar solución a los objetivos planteados para crear un sistema robusto y eficaz en el que confía el cliente.

4.1. Solución inicial

Podemos observar en el siguiente plano un esquema del conjunto general de la instalación. Los apartados abordados para dar con la solución inicial han sido el transporte del producto desde la máquina 1, el transporte del producto desde la máquina 2, la alimentación y transporte de palets, la mesa de paletizado y salida para almacenaje, la herramienta de cogida, y el sistema robot encargado de paletizar los productos.

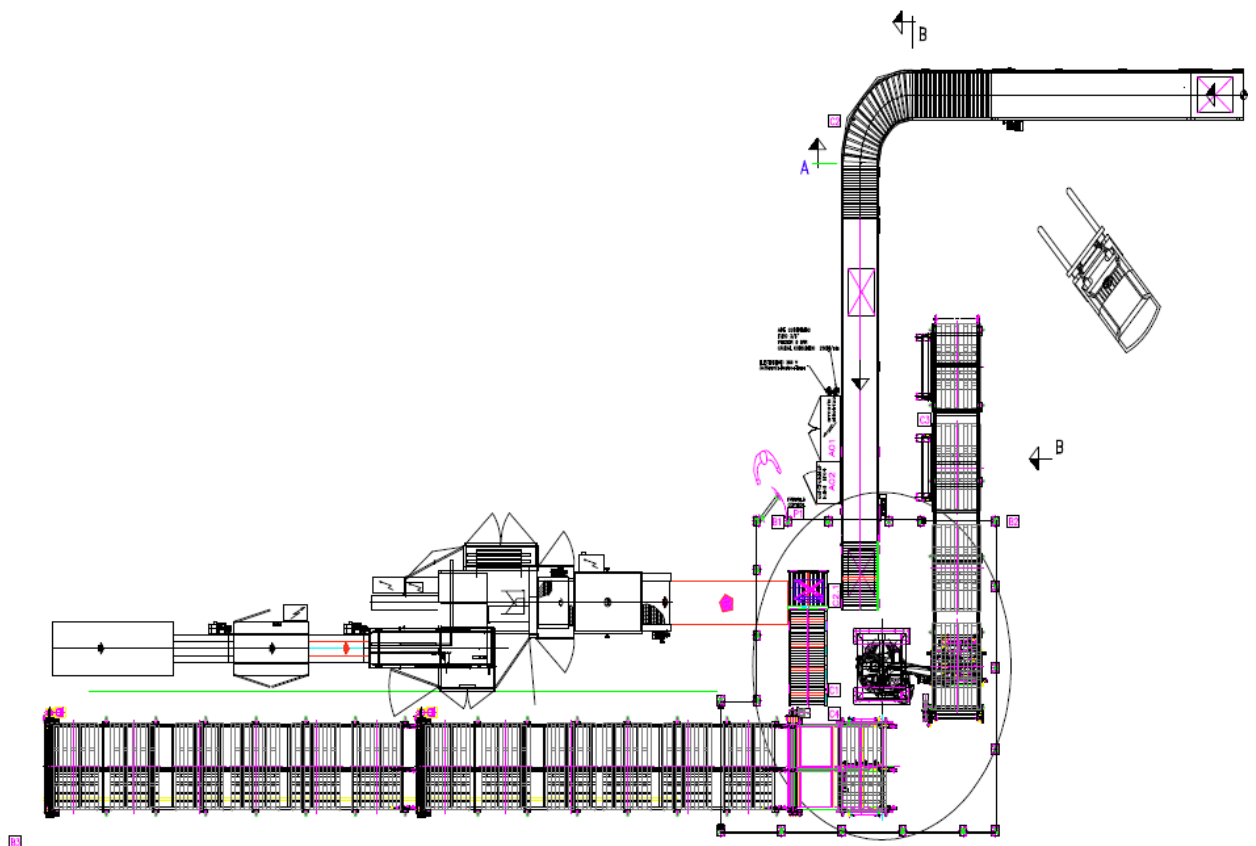


Ilustración 8 Solución inicial

4.1.1. Transporte máquina 1

El transporte del producto desde la salida de la máquina 1 hasta el punto de cogida del robot se ha llevado a cabo mediante 5 mesas de rodillos motorizados, diversos empujadores y fotocélulas para el control del transporte.

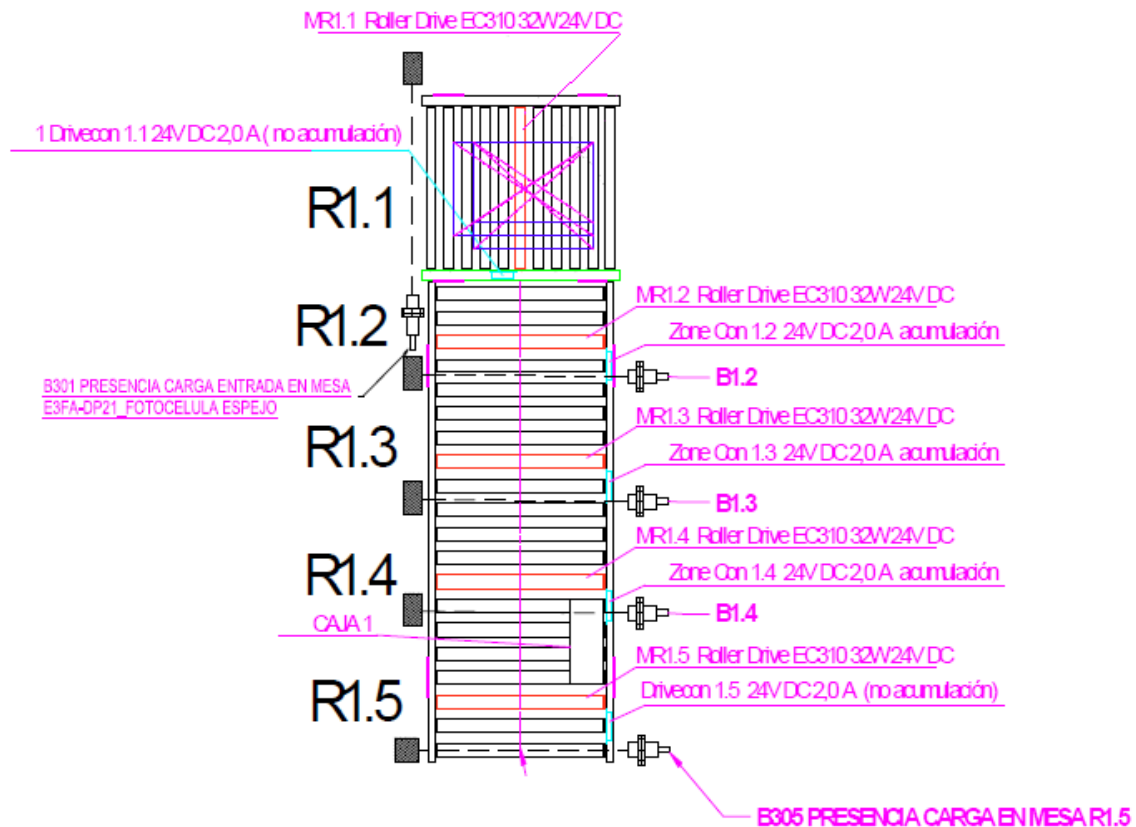


Ilustración 9 Transporte línea 1

La primera mesa y la quinta (R1.1 y R1.5 respectivamente) cuentan con un controlador **DriveControl** en el que las señales de marcha y paro se gobiernan desde el PLC. Las mesas R1.2, R1.3 y R1.4 se controlan mediante un **ZoneControl**, que conectados mediante un cable RJ-45 son capaces de transportar la carga de forma autónoma mediante acumulación sin presión.

El producto avanza hasta mesa R1.1 que se pondrá en marcha para acompañar el movimiento. Una vez el producto está sobre la mesa R1.1, un empujador lo lleva hasta la mesa R1.2 cuyos rodillos son perpendiculares a los de la mesa anterior. Una vez allí, el producto es transportado autónomamente hasta la mesa R1.5. Paramos el avance el producto mediante una fotocélula al final del trayecto y activamos un empujador, que nos separa los productos del borde de la mesa a la misma distancia para que sean cogidos por el robot.

4.1.2. Transporte máquina 2

Para el transporte del producto de la máquina 2 a su punto de cogida hemos empleado dos cintas transportadoras de banda, 6 mesas de rodillos motorizados y fotocélulas para el control del transporte. Podemos ver la distribución en el siguiente esquema:

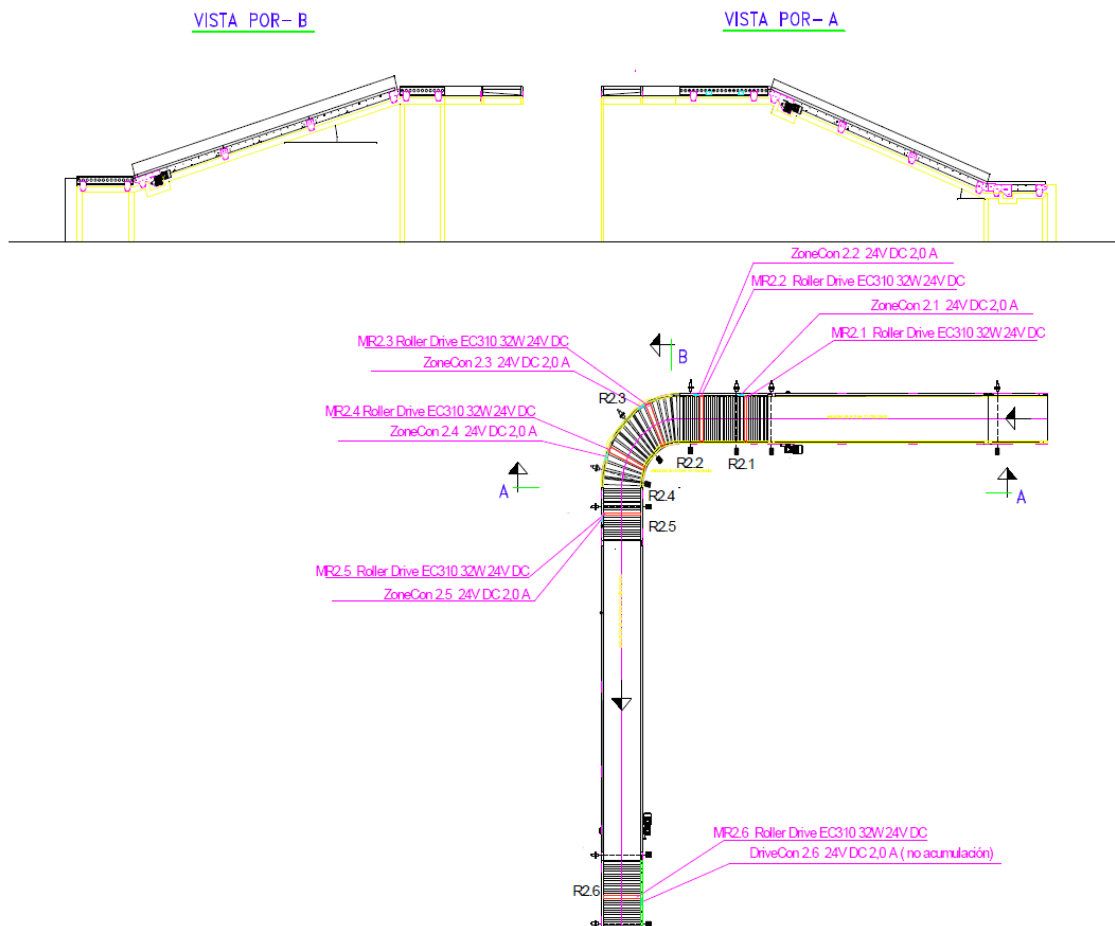


Ilustración 10 Transporte línea 2

Los productos entran en la cinta de banda 1 y avanzan hasta las mesas R2.1, R2.2, R2.3, R2.4 y R2.5. Estas mesas cuentan cada una con un controlador **ZoneControl**, por lo que realizan un transporte de acumulación sin presión de forma autónoma. Tras estas mesas, el producto entra en la cinta de banda 2 para acabar en la mesa de rodillos R2.6. En esta mesa detenemos el producto con una fotocélula para su cogida por parte del robot.

A lo largo de la cinta de banda 2 hay ancladas a cada lado dos guías de plástico que dirigen el producto de tal manera que entre centrado en la mesa de cogida R2.6.

4.1.3. Alimentador de palets

Para introducir los palets al sistema se ha diseñado una mesa de cadenas que es lo suficientemente grande para que el sistema no se detenga a la espera de un nuevo palet. Este cuenta con diversas fotocélulas para el control de la carga y seguridad, y un sensor de presencia para la carretilla elevadora

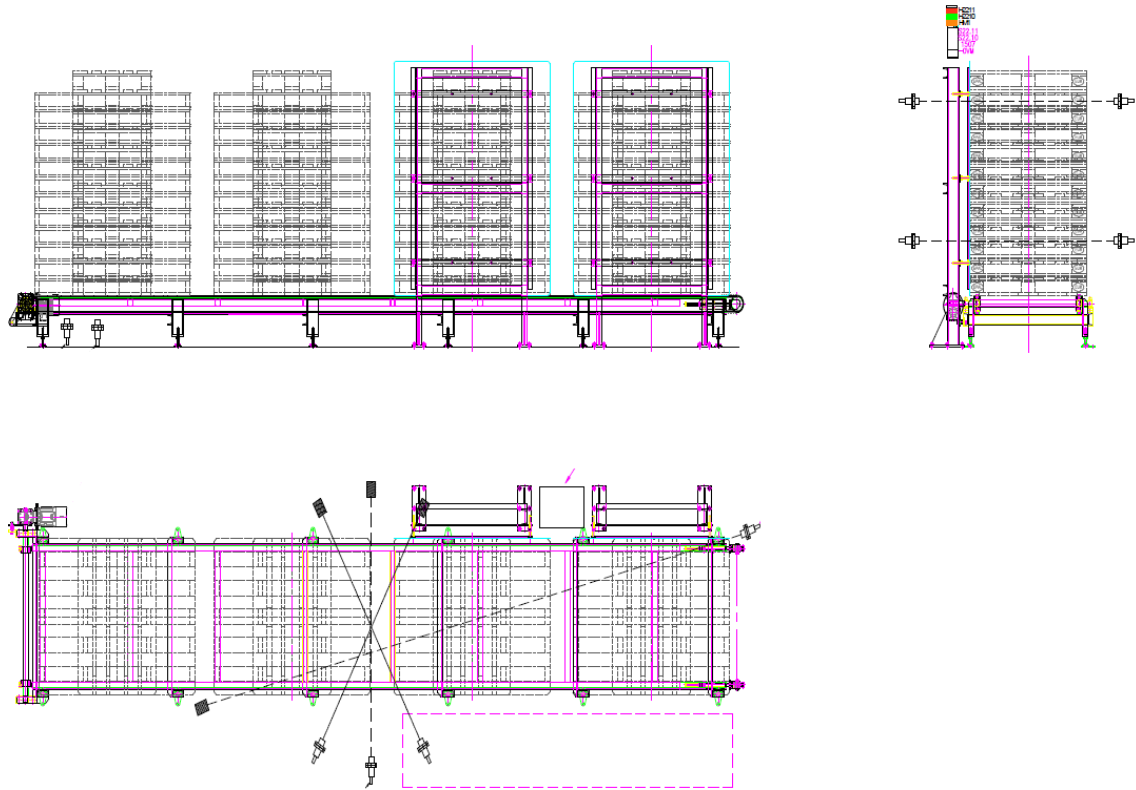


Ilustración 11 Alimentador palets

Las torres de palets se van a cargar en la mesa con una carretilla elevadora en la zona habilitada para ello. Esta zona está controlada con un sensor de presencia magnético que parará la mesa, si esta se encuentra en movimiento, para poder introducir los palets con seguridad. Cuando una torre de palets llegue al final de la mesa y alcance la primera fotocélula, reducirá su velocidad de avance para pararse del todo cuando alcance la última fotocélula. También cuenta con una fotocélula en lo alto que actúa de galibo y limita la altura máxima de las torres para que el robot pueda coger los palets sin problemas.

4.1.4. Mesa de paletizado y almacenaje

En este apartado de la instalación se ha optado por una mesa de cadenas más corta para la mesa de paletizado, y otras dos mesas de cadenas mucho más largas para dar salida a los palets con carga que van a ser almacenados. Estas dos mesas deben servir como pulmón para poder ir acumulando hasta que se pretenda almacenar los palets cargados.

La mesa de paletizado cuenta con unos centradores, empujados por electroválvulas neumáticas, encargados de colocar el palet en su correcta posición y sujetarlo mientras se paletiza el producto. Así mismo cuenta con dos fotocélulas para controlar la presencia de palet sobre la mesa y para sacar un palet cargado hacia la siguiente mesa.

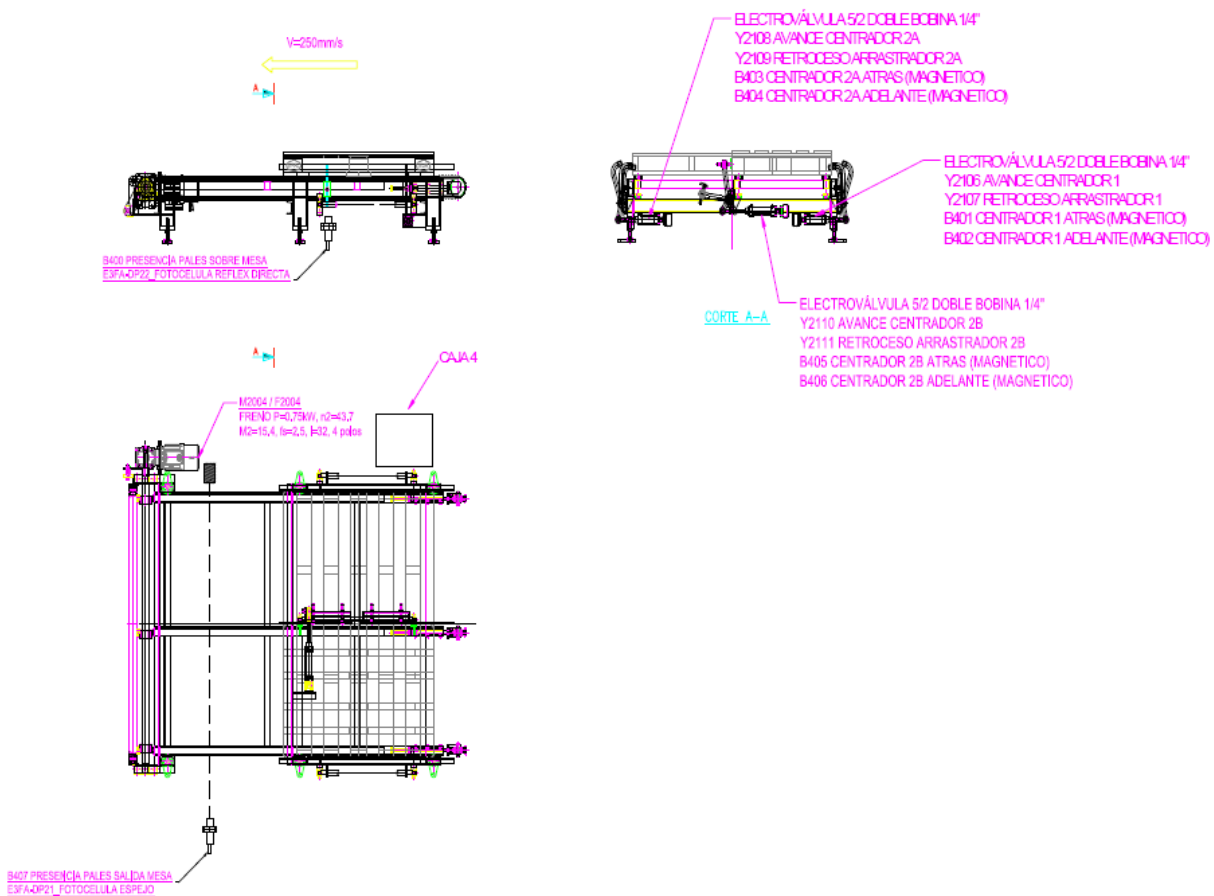


Ilustración 12 Mesa paletizado

Las mesas de salida para almacenaje cuentan con diversas fotocélulas para el control de los palets dentro de la mesa y el paso de una mesa a otra.

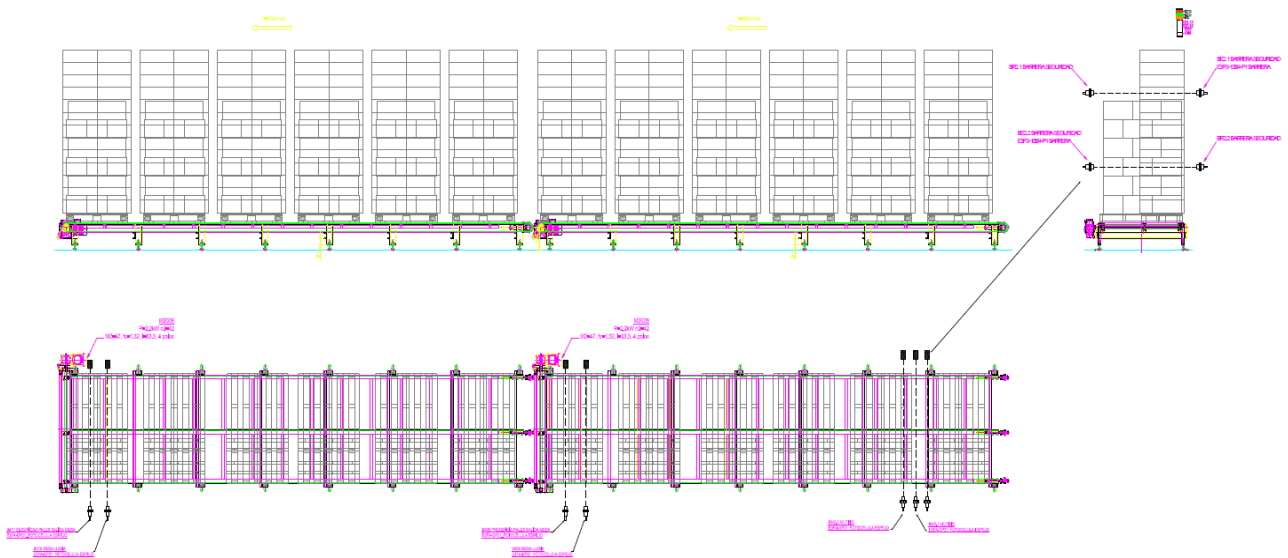


Ilustración 13 Línea almacenaje

El comportamiento de esta zona será el siguiente: cuando se termine un palet en la mesa de paletizado, este pasará a la primera mesa y así sucesivamente hasta que se llene la primera mesa. Una vez llena la primera mesa pasaremos toda la carga a la segunda mesa y procederemos de la misma manera. Este control lógico puede verse afectado si el operario requiere vaciar las mesas, que se irán acumulando al final de la segunda mesa para que se lleven los palets.

4.1.5. Herramienta cogida

La herramienta de cogida es el elemento con mayor importancia del sistema, puesto que tiene que ser capaz de coger todos los formatos de paquetes y a su vez coger los dos tipos de palets. Todo ello en el menor espacio posible para mejorar su maniobrabilidad.

La herramienta va a contar con dos partes: la garra, encargada de coger los palets; y la pinza, encargada de coger los productos.

La garra consiste en unos brazos de acero con uñas que cuando se cierran permiten sujetar un palet para su transporte. Los brazos se cierran por medio de dos cilindros neumáticos gobernados por una válvula de dos posiciones y dos detectores magnéticos para saber el estado en el que se encuentra.

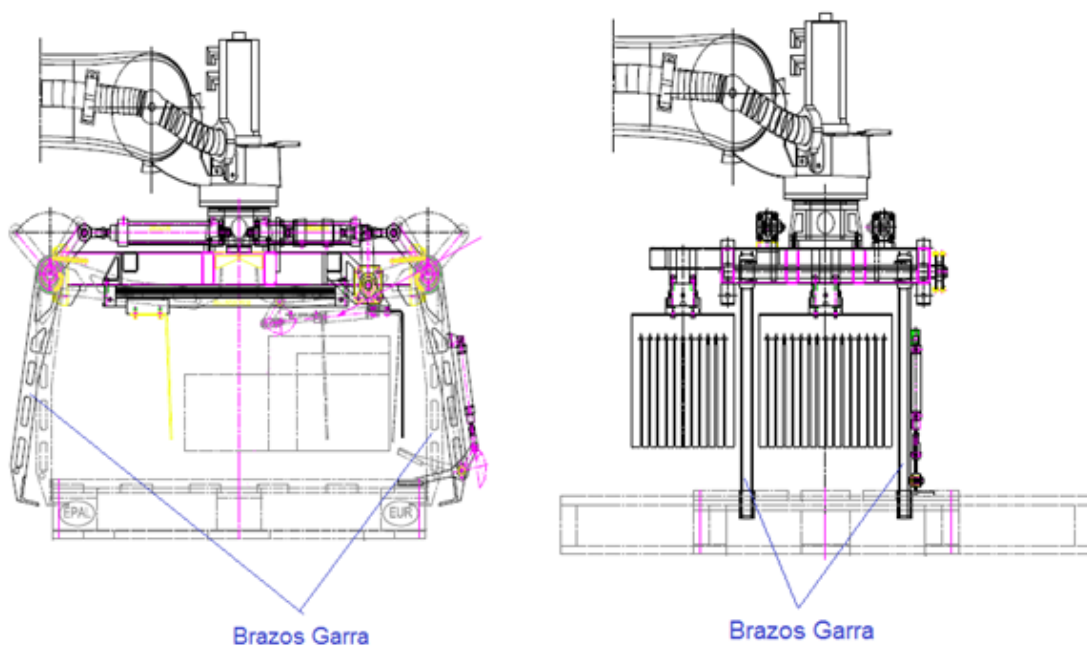


Ilustración 14 Brazos garra

Acoplado a uno de los brazos está el palpador de palets, un mecanismo que, mediante el uso de dos detectores magnéticos, nos permite saber cuándo la garra ha encontrado un palet. Este dispositivo cuenta con un cilindro neumático con una válvula de dos posiciones y sus respectivos detectores magnéticos para saber si está extendido o recogido.

Estando en posición extendida y mandándole la orden de extender el

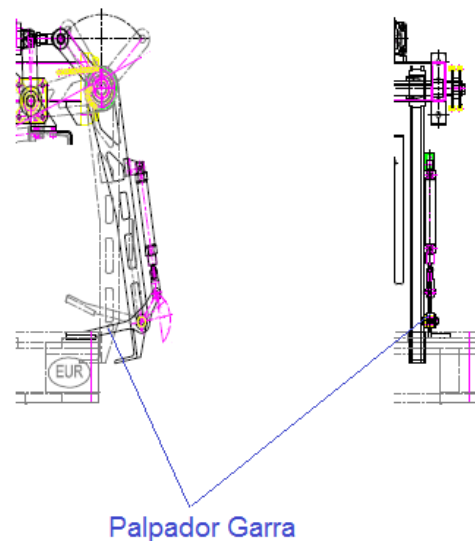


Ilustración 15 Palpador garra

palpador, si pierde la señal del detector de extendido, suponemos que ha palpado un palet.

La pinza que va a coger los fardos de rollos de papel consiste en dos palas, una fija y otra móvil. Cada una de las palas está dividida en dos partes, una más grande que otra, para que puedan cerrarse los brazos encargados de coger los palets. La pala fija es una plancha de acero compacta, mientras que la móvil es una plancha de acero a tiras diseñada para amoldarse al producto.

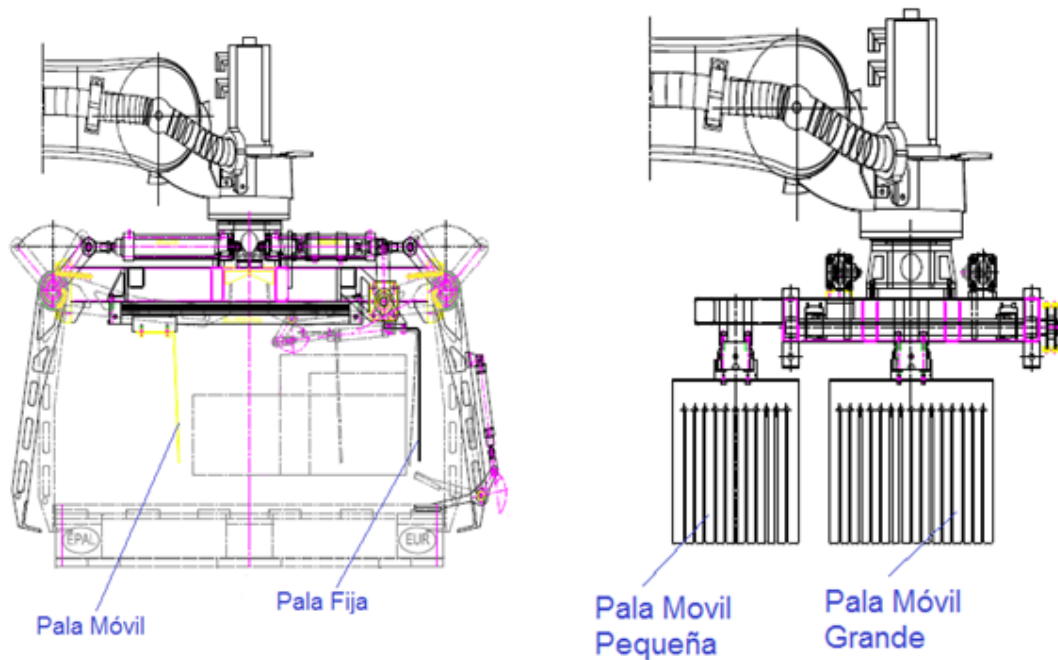


Ilustración 16 Palas de la pinza

La pala móvil se mueve mediante dos cilindros neumáticos controlados por dos electroválvulas, un cilindro para la parte grande de la pala y otro para la pequeña. Se ha separado el movimiento debido a que para coger los productos de la máquina 1 sólo se va a emplear la pala grande, mientras que para los de la máquina 2 utilizamos las dos palas. Así mismo, la pala móvil cuenta con un cilindro neumático y una válvula para darle cierta inclinación a la pala.

La presión que con la que cierran las palas móviles se regula mediante un regulador electrónico de presión. Este regulador nos permite pinzar los productos con la presión justa para que no se caigan ni sufran deformaciones por una presión excesiva.

4.1.6. Control del sistema

Todos y cada uno de los elementos que componen el sistema van a estar gobernados mediante un PLC industrial, en concreto el modelo **CJ2M** de la marca **OMRON** con puerto Ethernet/IP incorporado. Este autómatas cuenta con las siguientes tarjetas añadidas a su rack:

- La respectiva fuente de alimentación 100-240 VAC/5 VDC 2.8 A/24 VDC 0.4 A / 14W.
- Una unidad de 16 entradas digitales 24 VDC PNP.
- Dos unidades de 32 entradas digitales 24 VDC PNP.
- Una unidad de 16 salidas digitales transistor 24 VDC PNP 0.5 A.
- Una unidad de 32 salidas digitales transistor 24 VDC PNP 0.3 A.



Ilustración 17 Autómatas PLC

Conectado a la red Ethernet/IP se encuentra la estación remota NX de la marca **OMRON** que va a controlar todos los mecanismos de la herramienta de cogida del robot. La unidad de acoplamiento Ethernet/IP NX cuenta con las siguientes tarjetas NX:

- Unidad de 2 entradas analógicas 4-20 mA.
- Unidad de 2 salidas analógicas 4-20 mA.
- Unidad de 16 entradas digitales 24 VDC PNP.
- Unidad de 16 salidas digitales transistor 24 VDC PNP.
- Tapa de terminación.



Ilustración 18 Estación remota NX

También se encuentran conectados a la red Ethernet/IP la pantalla táctil **GP-4501TW** de la marca **PRO-FACE** y el controlador del sistema robot **KRC4** de la marca **KUKA**.

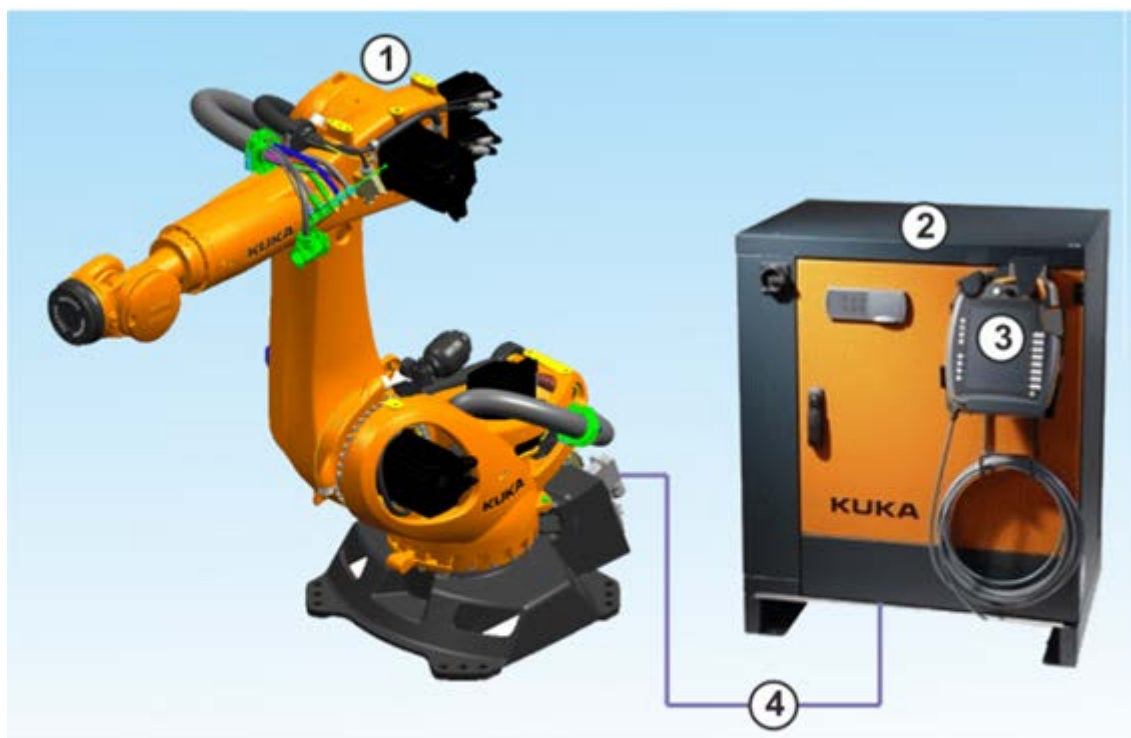
5. Robot industrial KUKA

Como el tema principal del presente Trabajo Fin de Grado es la realización del programa del robot dentro del sistema previamente descrito, se va a proceder a hacer una descripción más extensa de los elementos que conforman el sistema robot.

También se dará una introducción a los principios de movimiento del robot y a la programación lógica del mismo.

5.1. Elementos de un robot industrial

El robot industrial **KUKA** consta de los siguientes elementos básicos: el manipulador o brazo robótico, en este caso un **KR 180 R3200 PA**; la unidad de control del robot o controlador **KR-C4**; la unidad manual de programación o **KUKA smartPAD**; y los cables de unión.



1 Manipulador

3 Unidad manual de programación

2 Unidad de control del robot

4 Cables de unión

Ilustración 19 Sistema robot

5.1.1. Manipulador KR 180 R3200 PA

Un manipulador está formado por los siguientes grupos constructivos principales: muñeca, brazo, brazo de oscilación, columna giratoria, base, compensación de peso e instalación eléctrica.

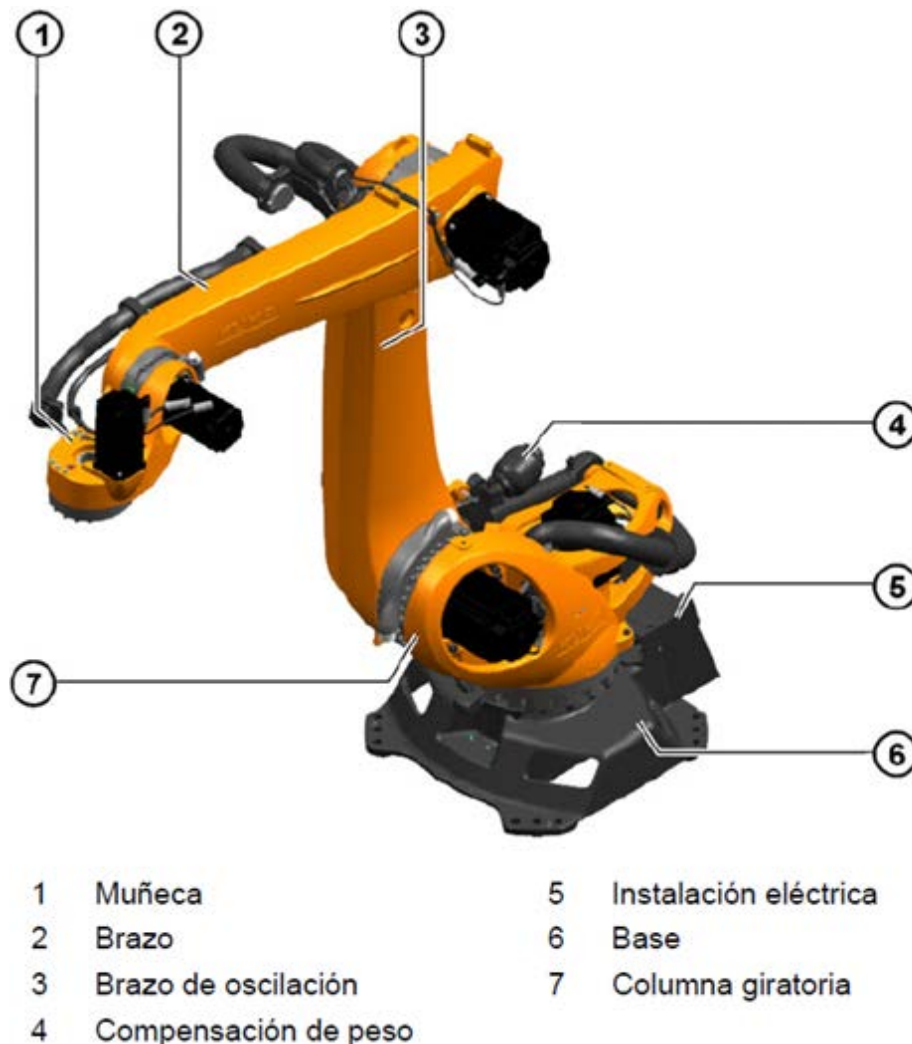


Ilustración 20 Brazo robotizado

- **Muñeca:** El robot **KR 180 R3200 PA** está equipado con una muñeca de dos ejes para cargas nominales de 180 kg. La muñeca está montada en el brazo por medio de un engranaje motor que la impulsa. Los componentes principales de la muñeca del eje hueco son el cuadro basculante, el motor del eje 6 y su correspondiente engranaje. Como accionamiento se utiliza un servomotor de CA sin escobillas con un freno monodisco de imán permanente y resolver del eje hueco, ambos integrados. En la brida de acople del eje 6 se pueden montar útiles o herramientas. La muñeca es de eje hueco y dispone de un taladro pasante con un diámetro de 60 mm.
- **Brazo:** El brazo es el elemento de transmisión entre la muñeca y el brazo de oscilación. Aloja el cuadro basculante de la muñeca a través del engranaje A5.

Esta combinación de engranaje y motor representa el eje 5, que no se puede controlar libremente durante el servicio. El accionamiento del brazo se efectúa con un servomotor de CA por medio del engranaje A3 montado entre el brazo y el brazo de oscilación. Este engranaje también actúa de alojamiento para el brazo. El motor del eje 3 está atornillado al brazo. El ángulo de giro máximo permitido está limitado mecánicamente por un tope en dirección positiva y negativa, los amortiguadores se encuentran montados en el brazo. Los topes correspondientes se encuentran en el brazo de oscilación.

- **Brazo de oscilación:** El brazo de oscilación es el grupo constructivo situado entre la columna giratoria y el brazo. Está montado en la columna giratoria, a un lado en el engranaje del eje 2, y está accionado por un servomotor de CA. En caso de movimientos alrededor del eje 2, el brazo de oscilación se mueve alrededor de la columna giratoria fija. El mazo de cables de la instalación eléctrica discurre por el interior del brazo de oscilación y está fijado con abrazaderas articuladas.
- **Columna giratoria:** La columna giratoria aloja los motores de los ejes 1 y 2. Los movimientos de giro del eje 1 se realizan mediante la columna giratoria. Está atornillada a la base a través del engranaje del eje 1. En el interior de la columna giratoria se encuentra el servomotor de CA que acciona el eje 1. En la parte posterior está integrado el contracojinete para la compensación de peso, en la carcasa de la columna giratoria.
- **Base:** La base es el soporte del robot. En la base se encuentran las interfaces de la instalación eléctrica y de las alimentaciones de energía. La base y la columna giratoria están unidas entre sí por el engranaje del eje 1.
- **Compensación de peso:** La compensación de peso es un grupo constructivo instalado entre la columna giratoria y el brazo de oscilación, que minimiza los movimientos generados alrededor del eje 2 durante la parada y el movimiento del robot. Para ello se utiliza un sistema hidroneumático cerrado. El sistema consta de dos acumuladores de presión, un cilindro hidráulico con los cables correspondientes y un manómetro.

Datos técnicos **KR 180 R3200 PA:**

- Cantidad de ejes: 5
- Volumen de campo de trabajo: 77,90 m³
- Repetibilidad de posición: $\pm 0,06$ mm
- Peso: 1093 kg
- Tipo de protección del robot y muñeca: IP 65

Datos de los ejes:

Eje	Zona de acción, limitada por software	Velocidad con la carga nominal
1	$\pm 185^\circ$	105 °/s
2	-5° a -140°	107 °/s
3	$+155^\circ$ a 0^{0*}	114 °/s
5	Eje no se puede seleccionar de forma activa	174 °/s
6	$\pm 350^\circ$	242 °/s

Tabla 3 Datos de los ejes

Direcciones de los movimientos y la asignación de cada eje.

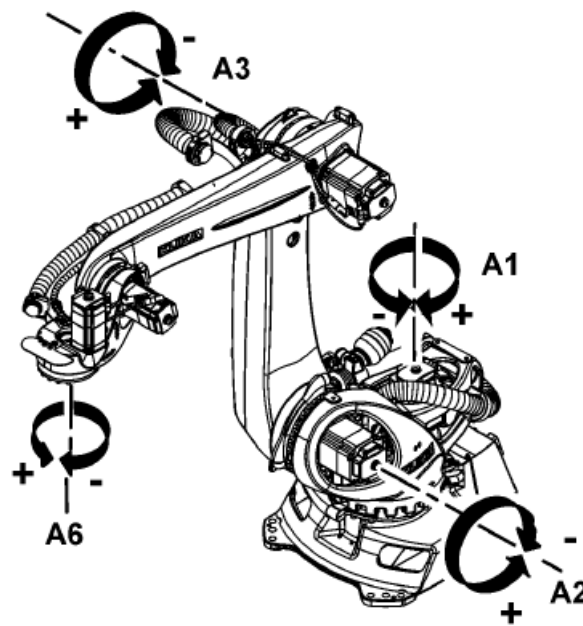


Ilustración 21 Direcciones ejes

Zona de trabajo:

La siguiente figura muestra el tamaño y la forma de las zonas de trabajo. El punto de referencia para la zona de trabajo es la intersección del eje 6 con la superficie de la brida de acople.

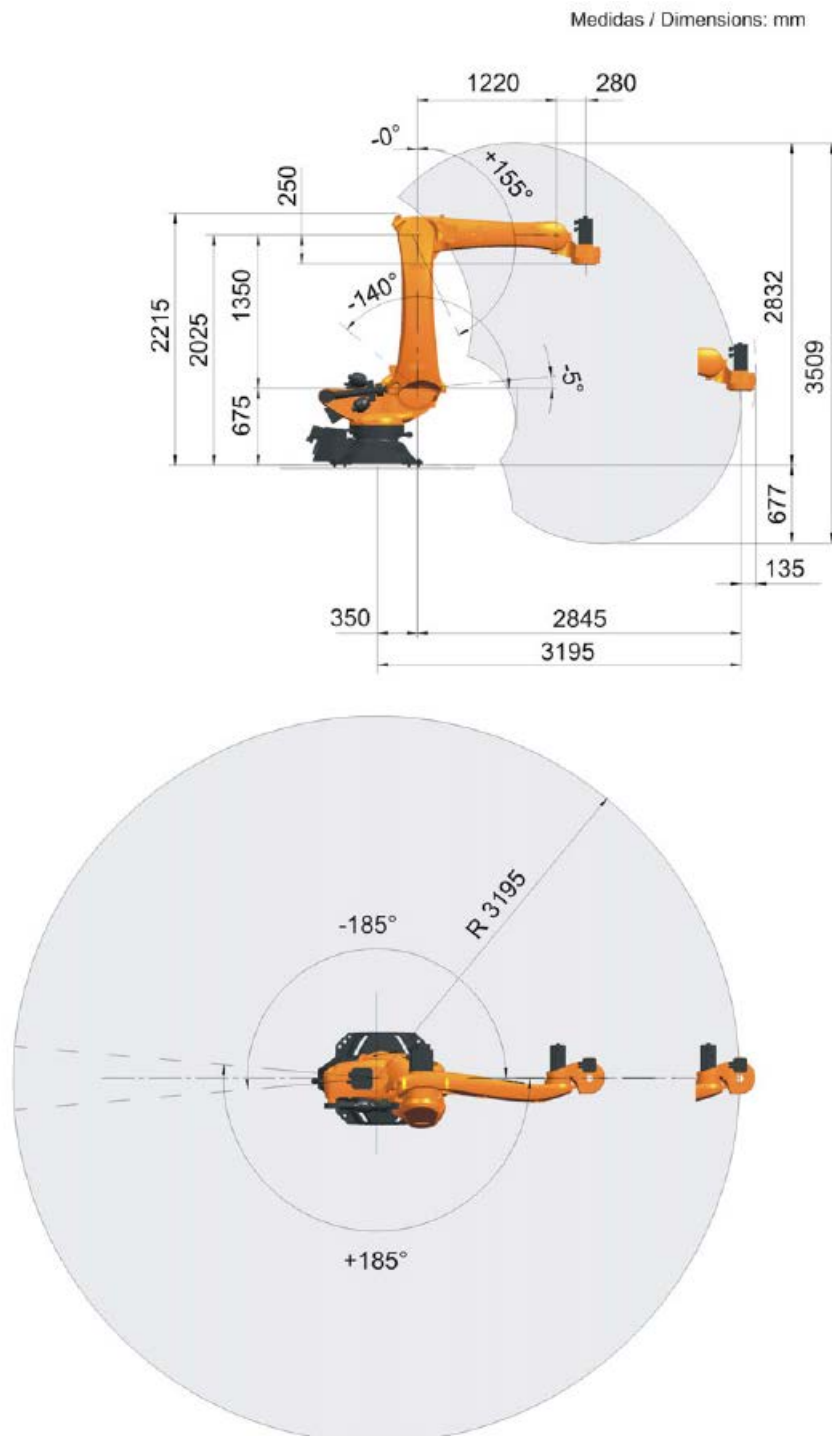


Ilustración 22 Zona de trabajo

5.1.2. Unidad de control del robot KR-C4

La unidad de control del robot es el componente principal en el que se integran los elementos de alimentación y control de los servomotores del robot KUKA. Posee las siguientes características:

- Peso: 150 kg
- Tipo de protección: IP 54
- Nivel de ruido: Valor medio 67 dB (A)
- Tensión nominal de conexión: AC 3x400 o AC 3x480 V (la unidad de control del robot solo puede conectarse a una red con punto de estrella puesto a tierra)
- Tolerancia admisible de la tensión nominal: Tensión nominal $\pm 10\%$
- Frecuencia de la red: 49 ... 61 Hz
- Temperatura ambiente en servicio sin equipo refrigeración: +5...+45°C (278...318 K)
- Temperatura ambiente en servicio con equipo refrigeración: +20...+20°C (293...323 K)



Ilustración 23 Unidad de control

Dentro del **KR-C4** se encuentran los siguientes elementos: PC de control (KPC), fuente de alimentación de baja tensión, fuente de alimentación del accionamiento con regulador del accionamiento KUKA Power-Pack, regulador del accionamiento KUKA Servo Pack (KSP), Cabinet Control Unit (CCU), Controller System Panel (CSP), Saefy Interface Board (SIB), Fusibles, Acumuladores, Ventilador, Panel de conexiones.

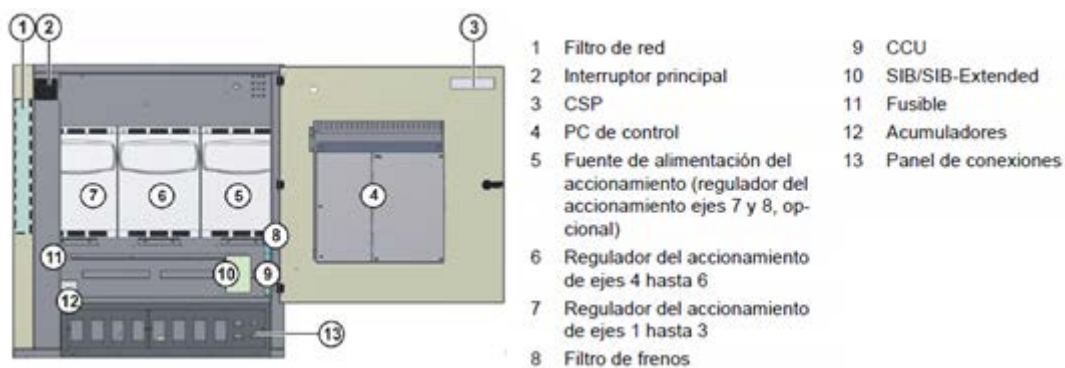


Ilustración 24 Elementos KRC4

- **KUKA Power Pack.** El KUKA Power Pack (KPP) es la fuente de alimentación del accionamiento y genera, a partir de una red de corriente trifásica, una tensión de circuito intermedio en el mismo sentido. Esta tensión de circuito intermedio suministra alimentación a los reguladores internos de accionamiento y a los accionamientos externos. Existen cuatro variantes distintas con el mismo

tamaño. El KPP contiene LED que indican el estado de servicio. El KPP tiene las siguientes funciones:

- Conexión y desconexión de la tensión de red.
- Alimentación de varios reforzadores de eje con circuitos intermediarios DC.
- Chopper de freno integrado con conexión a una resistencia de lastre.
- Control de sobrecarga de la resistencia de lastre.
- Parada de los servomotores sincronizados mediante el frenado por cortocircuito.

- **KUKA Servo Pack:** El KUKA Servo Pack (KSP) es el regulador del accionamiento de los ejes del manipulador. Existen 3 variantes distintas con el mismo tamaño. El KSP contiene LED que indican el estado de servicio. El KSP tiene las siguientes funciones:

- Alimentación directa de la tensión de circuito intermedio DC.
- Regulación de corriente por orientación de campo para servomotores: Regulación de momento de giro.

- **PC de control:** El PC de control (KPC) contiene los siguientes componentes:

- Fuente de alimentación.
- Placa base.
- Procesador.
- Disipador de calor.
- Motores de almacenamiento.
- Disco duro.
- Tarjeta de red LAN-Dual-NIC.
- Ventilador del PC.
- Grupos constructivos opcionales: tarjeta de bus de campo.

El PC de control (KPC) asume las siguientes funciones de la unidad de control de robot:

- Superficie de usuario.
- Realización, corrección, archivado y mantenimiento de programas.
- Control de proceso.
- Proyecto de trayectoria.
- Mando del circuito de accionamientos cartesiano.
- Técnica de seguridad.
- Comunicación con la periferia externa (otras unidades de control, ordenador de gestión superior. PCs, red).

- **Cabinet Control Unit:** La Cabinet Control Unit (CCU) es la central de distribución de corriente y la interfaz de comunicación para todos los componentes de la unidad de control de robot. La CCU consta de una Cabinet Interface Board (CIB)

y de una Power Management Board (PMB). Todos los datos se transmiten a través de la comunicación interna a la unidad de control, donde se continúan tratando. En caso de fallo de la tensión de red unos acumuladores se encargan de suministrar corriente a la unidad de control hasta que se han guardado los datos de posición y se ha parado la unidad de control. Por medio de una prueba de carga se comprueba el estado de carga y la calidad de los acumuladores.

- **Saefty Interface Board:** La Saefty Interface Board (SIB) forma parte de la interfaz segura del cliente. En función de la estructura de la interfaz del cliente, la unidad de control del robot utiliza dos SIB distintas: la placa de la SIB estándar y la de la SIB Extended. Ambas placas se pueden hacer funcionar solas o juntas. Tanto la SIB estándar como la Extended poseen fundamentalmente funciones de registro, control y conmutación. Para las señales de salida se dispone de salidas separadas galvánicamente.
- **Convertidor Resolver Digital:** con el Resolver Digital Converter (RDC) se registran datos de posición del motor. Se pueden conectar 8 resolveres al RDC. Además, se mide y evalúa la temperatura del motor. Para guardar datos no volátiles en la caja RDC se halla la EDS. El RDC está fijado, dentro de una caja RDC, a la base del manipulador.
- **Controller System Panel:** el Controller System Panel (CSP) es un indicador que muestra el estado de servicio y tiene las conexiones siguientes: USB1, USB2, KLI(opcional).

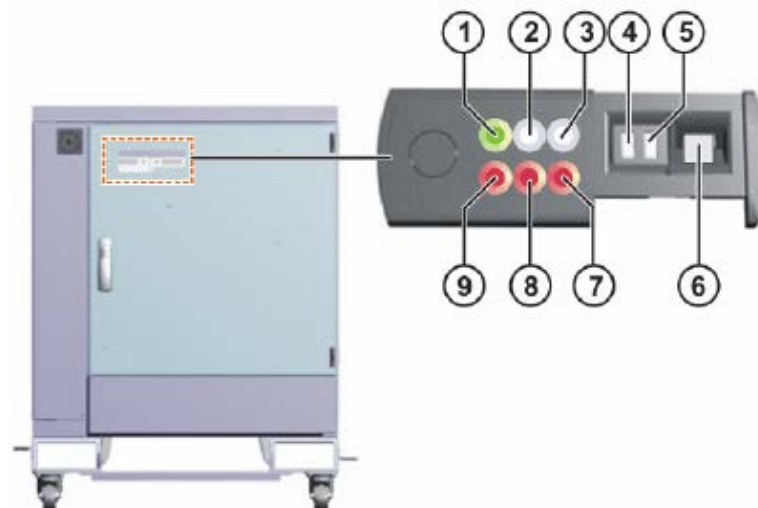


Ilustración 25 Frontal del KRC4

Pos.	Pieza	Color	Significado
1	LED 1	Verde	LED modo de servicio
2	LED 2	Blanco	LED Sleep
3	LED 3	Blanco	LED automático
4	USB 1	-	-
5	USB 2	-	-
6	RJ45	-	KLI
7	LED 6	Rojo	LED fallo 3
8	LED 5	Rojo	LED fallo 2
9	LED 4	Rojo	LED fallo 1

Tabla 4 Frontal del KRC4

- **Fuente de alimentación de baja tensión:** la fuente de alimentación de baja tensión suministra tensión a los componentes de la unidad de control del robot. Un LED verde muestra el estado de servicio de la fuente de alimentación de baja tensión.
- **Acumuladores:** en caso de fallo de la red o de desconexión de la corriente, unos acumuladores se encargan de apagar la unidad de control del robot de modo controlado. La CCU carga estos acumuladores y constantemente se comprueba y muestra su estado de carga.
- **Filtro de red:** suprime las posibles tensiones de perturbación del cable de red.

5.1.3. Unidad manual de programación KUKA smartPAD

El **smartPAD** es la unidad manual de programación del robot industrial. El **smartPAD** contiene todas las funciones de control e indicación necesarias para el manejo y la programación del robot industrial. Dispone de una pantalla táctil que se puede manejar con un dedo o un lápiz. No es necesario utilizar un ratón o teclado externo.

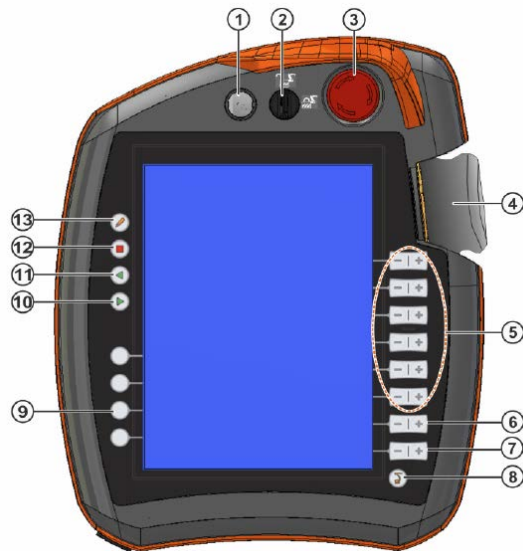


Ilustración 26 Vista delantera smartPAD



Ilustración 27 Vista trasera smartPAD

Pos.	Descripción
1	Botón para desenchufar el smartPAD
2	Interruptor de llave para acceder al gestor de conexiones. El conmutador únicamente se puede cambiar cuando está insertada la llave. El gestor de conexiones permite cambiar el modo de servicio.
3	Dispositivo de PARADA DE EMERGENCIA. Para detener el robot en situaciones de peligro. El dispositivo de PARADA DE EMERGENCIA se bloquea cuando se acciona.
4	Space Mouse: para el desplazamiento manual del robot.
5	Teclas de desplazamiento: para el desplazamiento manual del robot.
6	Tecla para ajustar el override de programa.
7	Tecla para ajustar el override manual.
8	Tecla de menú principal: muestra las opciones de menú en el smartHMI.
9	Teclas de estado. Las teclas de estado sirven principalmente para ajustar los parámetros de paquetes tecnológicos. Su función exacta depende del paquete tecnológico instalado.
10	Tecla de arranque: con la tecla de arranque se inicia un programa.
11	Tecla de arranque hacia atrás: con la tecla de arranque hacia atrás se inicia un programa en sentido inverso. El programa se ejecuta paso a paso.
12	Tecla STOP: con la tecla STOP se detiene un programa en ejecución.
13	Tecla del teclado: Muestra el teclado. Generalmente no es necesario mostrar el teclado porque el smartHMI detecta cuándo es necesario introducir datos con el teclado y lo abre automáticamente.

Tabla 6 Vista delantera smartPAD

1	Pulsador de hombre muerto	4	Conexión USB
2	Tecla de arranque (verde)	5	Pulsador de hombre muerto
3	Pulsador de hombre muerto	6	Placa de características

Tabla 5 Vista trasera smartPAD

El selector de modo de servicio conmuta los cuatro modos de funcionamiento del robot:

T1, T2, Automático y Automático Externo.



En los modos de funcionamiento **T1** y **T2** se ejecutan movimientos paso a paso del programa manteniendo el pulsador de hombre muerto y tecla de arranque. Una vez realizada una trayectoria el robot se detiene siendo necesario pulsar de nuevo la tecla de arranque. En el modo

T1 la velocidad máxima de movimiento está limitada a 250mm/s y puede moverse el robot manualmente. En el modo **T2** la velocidad no está limitada y se puede ejecutar movimientos a la velocidad de automático.

Para ejecutar un programa en el modo Automático es necesario seleccionar dicho programa desde la consola y pulsar la tecla de arranque a la velocidad programada. En el modo Automático Externo la selección y marcha del programa se ejecuta por medio de una unidad de control superior como puede ser un PLC. Es necesario que el circuito de seguridad del robot esté cerrado para poder moverse en automático.



La seta de emergencia del **smartPAD** debe pulsarse en situaciones de emergencia. El manipulador y los ejes adicionales se detienen, se desconectan los accionamientos y la alimentación de tensión de los frenos. Para poder seguir con el servicio, debe desenclavarse el dispositivo de **PARADA DE EMERGENCIA** girándolo.

La tecla de arranque inicia un programa cuando se hayan activado los accionamientos en el modo de servicio automático. En los modos de servicio **T1** y **T2** se ejecuta un programa paso a paso hacia delante mientras se mantenga la tecla y el pulsador de hombre muerto. Pulsando la tecla de arranque hacia atrás se ejecuta un programa paso a paso hacia atrás del mismo modo antes descrito. La tecla STOP detiene la ejecución del programa en automático, efectuando una parada sobre la trayectoria.

En el modo T1 podemos mover manualmente el robot mediante las teclas de desplazamiento o el **Space Mouse**. Si bien vamos a moverlo con las teclas de desplazamiento debemos elegir si queremos desplazar eje a eje o de forma cartesiana. Para utilizar el **Space Mouse** (recomendado para moverse en el sistema de coordenadas del robot) debemos definir nuestra orientación respecto al robot y los ajustes de desplazamiento.



Ilustración 28 Configuración Space Mouse

El **pulsador de hombre muerto** tiene 3 posiciones: no pulsado, posición intermedia o pulsado a fondo. En los modos de servicio T1 y T2, el pulsador de hombre muerto debe mantenerse en la posición intermedia para poder efectuar movimientos con el manipulador. En los modos de servicio Automático y Automático Externo, el pulsador de hombre muerto carece de función.

5.2. Sistema de coordenadas

Para indicar en el espacio la posición y orientación de los puntos del robot necesitamos emplear sistemas de coordenadas. En la unidad de control de robot se encuentra definidos los siguientes sistemas de coordenadas cartesianos: **WORLD**, **ROBROOT**, **BASE** Y **TOOL**.

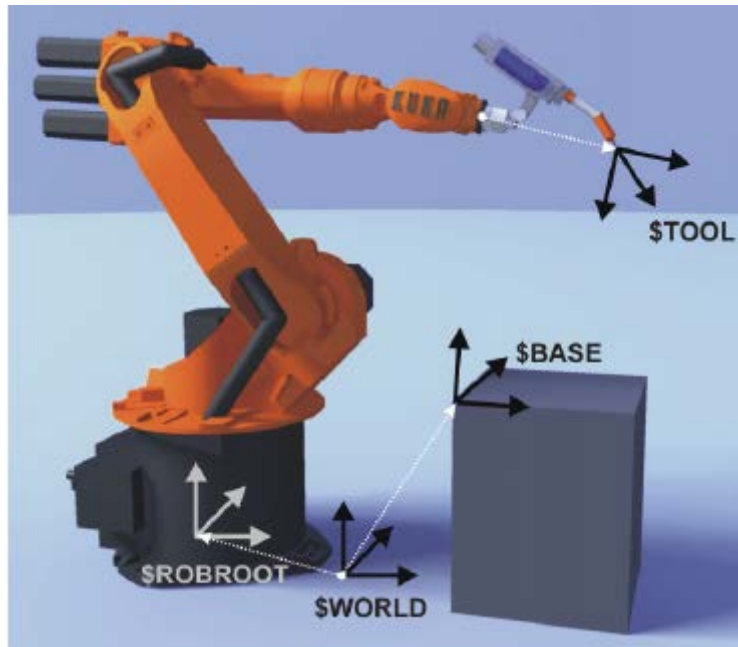


Ilustración 29 Sistemas de coordenadas

5.2.1. Sistema de coordenadas WORLD

El sistema de coordenadas **WORLD** (sistema de coordenadas universales) es un sistema de coordenadas cartesianas de definición fija. Es el sistema genérico de coordenadas para los sistemas de coordenadas **BASE** y **ROBROOT**.

Por defecto, el sistema de coordenadas **WORLD** se encuentra en el pie del robot.



Ilustración 30 Sistema WORLD

5.2.2. Sistema de coordenadas ROBROOT

El sistema de coordenadas **ROBROOT** es un sistema de coordenadas cartesianas que siempre se encuentra en el pie del robot. Describe la posición de robot en relación al sistema de coordenadas **WORLD**.

Por defecto, el sistema de coordenadas **ROBROOT** se cubre con el sistema de coordenadas **WORLD**. Con **\$ROBROOT** (variable interna del sistema) puede definirse un corrimiento del robot respecto al sistema de coordenadas **WORLD**.

5.2.3. Sistema de coordenadas BASE

El sistema de coordenadas **BASE** es un sistema de coordenadas cartesianas que describe la posición de la pieza de trabajo. Hace referencia al sistema de coordenadas **WORLD**.

Por defecto, el sistema de coordenadas **BASE** se cubre con el sistema de coordenadas **WORLD**. Es desplazado por el usuario hacia la pieza de trabajo. Podemos guardar hasta 32 sistemas de coordenadas **BASE**.

5.2.3.1. Medir la base

En la medición de la base se asigna una superficie de trabajo a un sistema de coordenadas cartesianas. El sistema de coordenadas **BASE** tiene su origen en un punto definido por el usuario.

Al crear una base el **TCP** puede moverse de forma manual a lo largo de la superficie de trabajo. Los puntos pueden ser programados por aprendizaje en relación a la base. Si la Base debe ser desplazada, p. ej. si la superficie de trabajo fue desplazada, se desplazan también los puntos y no tienen que ser programados nuevamente.

Existen tres métodos para crear una base:

- **Método de los tres puntos:** es el método más empleado. Se situará el **TCP** manualmente en el punto de origen del nuevo sistema de coordenadas, guardamos ese punto. Ahora movemos el **TCP** en la dirección X positiva del nuevo sistema, guardamos el punto. Finalmente desplazamos el **TCP** hacia un Y positivo del plano XY del nuevo sistema y terminamos la medición.
- **Método indirecto:** es utilizado cuando no es posible llegar con el robot al origen de la base, p. ej. porque se encuentra en el interior de una pieza o fuera del campo de trabajo del robot.
Debe efectuarse el desplazamiento a cuatro puntos de la base, cuyas coordenadas deben conocerse. La unidad de control del robot calcula la base utilizando estos puntos.
- **Entrada numérica:** podemos introducir numéricamente los valores de distancia del origen de la base al origen del sistema de coordenadas **WORLD** y los valores de giro de los ejes de la base en función del sistema de coordenadas **WORLD**.

5.2.4. Sistema de coordenadas TOOL

El sistema de coordenadas **TOOL** es un simple sistema de coordenadas cartesianas cuyo punto de trabajo se encuentra en la herramienta.

Por defecto, el origen del sistema de coordenadas **TOOL** se encuentra en el centro de la brida. El sistema de coordenadas **TOOL** es desplazado por el usuario en el punto de trabajo de la herramienta. Podemos guardar hasta 16 sistemas de coordenadas **TOOL**.

5.2.4.1. Medir la herramienta

En la medición de la herramienta, el usuario asigna a una herramienta montada en la brida de acople del robot, un sistema de coordenadas cartesianas **TOOL**.

El sistema de coordenadas **TOOL** tiene su origen en un punto definido por el usuario. Este se denomina **TCP** (Tool Center Point). Por regla general, el **TCP** se coloca en el punto de trabajo de la herramienta.

Al tratarse de un robot de paletizado, el método de medición es la entrada numérica de las medidas de la herramienta que está montada sobre la brida.

5.2.5. Movimientos manuales

Existen dos elementos de operación con los cuales puede moverse un robot: **Teclas de desplazamiento y Space Mouse**.

El override manual determina la velocidad del robot en el desplazamiento en modo manual. La velocidad que el robot alcanza realmente con el 100% del override no debe superar 250mm/s. Podemos ajustar este parámetro mediante la tecla positiva negativa del smartPAD en los pasos 100%, 75%, 50%, 30%, 10%, 3%, 1%; o bien mediante el indicador de estado POV/HOV a pasos de 1%.

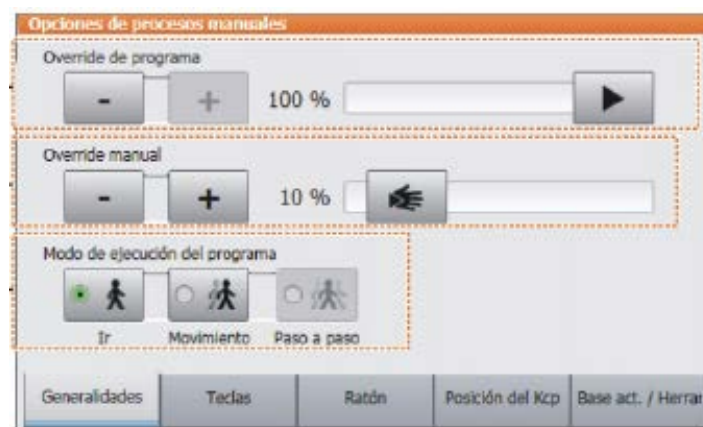


Ilustración 31 Movimientos manuales

Existen dos formas de desplazar el robot de forma manual:

- **Desplazamiento cartesiano:** el TCP es desplazado en dirección positiva o negativa a lo largo de los ejes de un sistema de coordenadas. Debemos seleccionar el sistema de coordenadas en el que nos vamos a mover (Mundo,

Base o Herramienta). Junto a las teclas de desplazamiento se visualizan los parámetros X, Y, Z, A, B, C para los movimientos lineales y de rotación. Para realizar el movimiento debemos mantener el pulsador de hombre muerto y pulsar las teclas de desplazamiento o mover el Space Mouse previamente configurado.

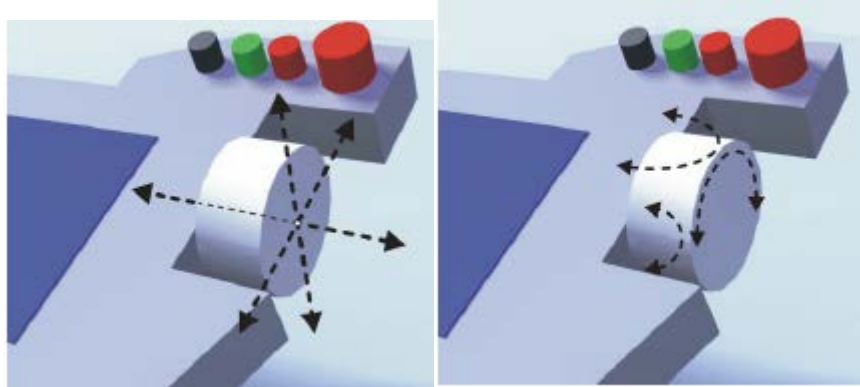


Ilustración 32 Movimientos Space Mouse

- **Desplazamiento específico de los ejes:** cada eje puede se puede desplazar individualmente en dirección positiva y negativa. Junto a las teclas de desplazamiento se indican los ejes disponibles para mover. Para realizar el movimiento basta con mantener el pulsador de hombre muerto y pulsar la tecla de desplazamiento.



Ilustración 33 Desplazamiento ejes

5.3. Movimientos del robot

Se pueden programar los siguientes tipos de movimiento:

- Movimiento punto a punto (PTP).
- Movimiento lineal (LIN).
- Movimiento circular (CIRC).
- Movimiento spline.

5.3.1. Posicionamiento aproximado

Posicionamiento aproximado significa que el robot no se detiene exactamente sobre el punto programado. El posicionamiento aproximado es una opción que puede seleccionarse en la programación de movimientos.

El posicionamiento aproximado no es posible cuando a la instrucción de movimiento le sigue una instrucción que provoca una parada del procesamiento.

5.3.1.1. Posicionamiento aproximado en un PTP

El **TCP** abandona la trayectoria sobre la cual se posicionaría exactamente en el punto de destino y recorre una trayectoria más rápida. En la programación del movimiento se define la distancia mínima al punto de destino en que el **TCP** puede abandonar su trayectoria original.

El transcurso de una trayectoria en un movimiento PTP con posicionamiento aproximado no es previsible. Tampoco es previsible por que lado del punto programado por aproximación pasa la trayectoria.



Ilustración 34 Aproximación PTP

5.3.1.2. Posicionamiento aproximado en un LIN y CIRC

El **TCP** abandona la trayectoria sobre la cual se posicionaría exactamente en el punto de destino y recorre una trayectoria más corta. En la programación del movimiento se define la distancia mínima al punto de destino en que el **TCP** puede abandonar su trayectoria original.

En un movimiento CIRC con posicionamiento aproximado se pasa exactamente por el punto auxiliar.



Ilustración 35 Aproximación LIN y CIRC

5.3.1.3. Posicionamiento aproximado en un Spline

Se puede realizar una aproximación conjunta de los bloques Spline y los movimientos individuales Spline. No importa si se trata de bloques Spline CP o PTP ni el tipo de movimiento individual.

El arco de aproximación se corresponde siempre con el segundo movimiento, en lo que respecta al tipo de movimiento. Durante la aproximación SPTP-SLIN, por ejemplo, el arco de aproximación es del tipo CP.

Los movimientos Spline no se pueden aproximar con movimientos convencionales (LIN, CIRC, PTP). La aproximación no es posible por tiempo o parada del movimiento de avance:

Si, por motivos de tiempo o por una parada del movimiento de avance, no es posible efectuar una aproximación, el robot espera al comienzo del arco de aproximación.

- Debido a motivos de tiempo: el robot continúa en el momento en que se pueda planear el paso siguiente.
- Debido a una parada del movimiento de avance: al iniciarse el arco de aproximación se alcanza el final del paso actual. Es decir, la parada del movimiento de avance se anula y se puede planear el siguiente paso. El robot se sigue desplazando.

En ambos casos el robot recorre el arco de aproximación. De hecho, la aproximación es posible, solo se retarda.

Este comportamiento es contrario a los movimientos LIN, CIRC o PTP. Si en estos casos, por los motivos anteriormente citados, no se puede efectuar una aproximación, el robot se desplaza exactamente hasta el punto de destino.

5.3.2. Tipo movimiento PTP

En el movimiento punto a punto el robot desplaza el **TCP** al punto de destino a lo largo de la trayectoria más rápida. La trayectoria más rápida no es, en regla general, la trayectoria más corta y por ello no es una recta. Dado que los ejes del robot se mueven de forma rotacional, trayectorias curvas pueden ser ejecutadas de forma más rápida que las rectas. No puede predecirse la trayectoria exacta.

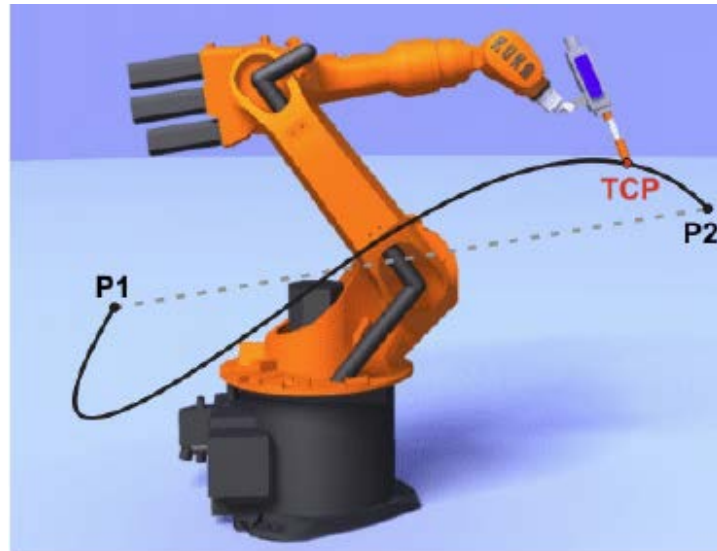


Ilustración 36 Movimiento PTP

5.3.2.1. Programación de movimiento PTP

Podemos programar un movimiento PTP mediante el formulario de la consola **Instrucciones > Movimiento > PTP**. Se nos despliega un formulario inline:

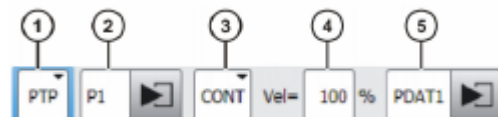


Ilustración 37 Formulario PTP

Pos.	Descripción
1	Tipo de movimiento PTP
2	Nombre del punto de destino El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.
3	■ CONT : El punto de destino es de posicionamiento aproximado. ■ [vacío] : El punto de destino se alcanza con exactitud.
4	Velocidad ■ 1 ... 100 %
5	Nombre para el paso de movimiento El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.

Tabla 7 Formulario PTP

5.3.3. Tipo movimiento LIN

En el movimiento lineal el robot conduce el **TCP** a la velocidad definida hasta el punto de destino a lo largo de una recta.

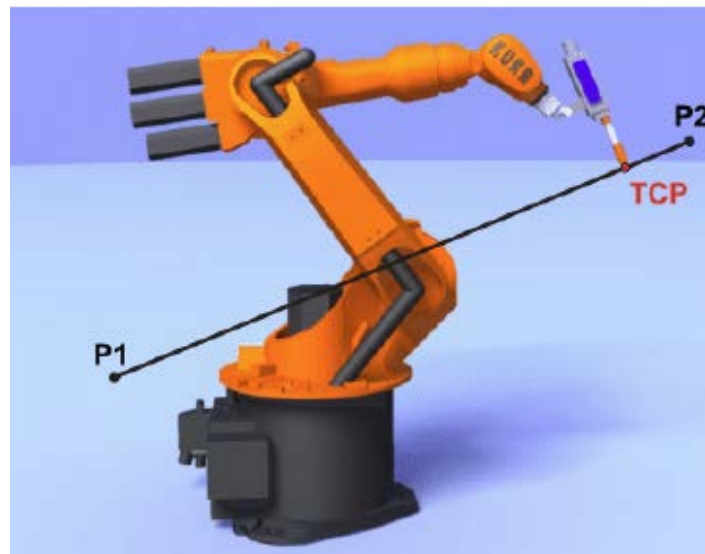


Ilustración 38 Movimiento LIN

5.3.3.1. Programación de movimiento LIN

Podemos programar un movimiento LIN mediante el formulario de la consola **Instrucciones > Movimiento > LIN**. Se nos despliega un formulario inline:

Ilustración 39 Formulario LIN

Pos.	Descripción
1	Tipo de movimiento LIN
2	Nombre del punto de destino El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.
3	■ CONT : El punto de destino es de posicionamiento aproximado. ■ [vacío]: El punto de destino se alcanza con exactitud.
4	Velocidad ■ 0.001 ... 2 m/s
5	Nombre para el paso de movimiento El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.

Tabla 8 Formulario LIN

5.3.4. Tipo de movimiento CIRC

En el movimiento circular el robot conduce el **TCP** con una velocidad definida al punto de destino a lo largo de una trayectoria circular. La trayectoria circular queda definida por el punto de arranque, un punto intermedio y el punto de destino.

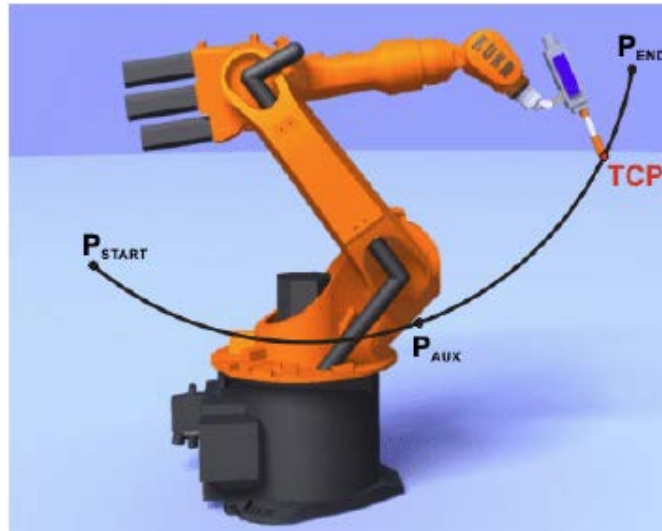


Ilustración 40 Movimiento CIRC

5.3.4.1. Programación de movimiento CIRC

Podemos programar un movimiento CIRC mediante el formulario inline. Primero desplazamos el **TCP** a la posición que se programará por aprendizaje como punto auxiliar, después seleccionamos la secuencia **Instrucciones > Movimiento > CIRC** y ajustamos los parámetros del formulario. Pulsamos TouchUP PI y desplazamos el **TCP** a la posición que se programa por aprendizaje como punto de destino.

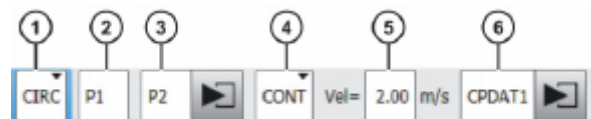


Ilustración 41 Formulario CIRC

Pos.	Descripción
1	Tipo de movimiento CIRC
2	Nombre del punto auxiliar El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.
3	Nombre del punto de destino El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.
4	<ul style="list-style-type: none"> ■ CONT: El punto de destino es de posicionamiento aproximado. ■ [vacío]: El punto de destino se alcanza con exactitud.
5	Velocidad ■ 0.001 ... 2 m/s
6	Nombre para el paso de movimiento El sistema asigna automáticamente un nombre. El nombre puede sobrescribirse.

Tabla 9 Formulario CIRC

5.3.5. Tipo de movimiento Spline

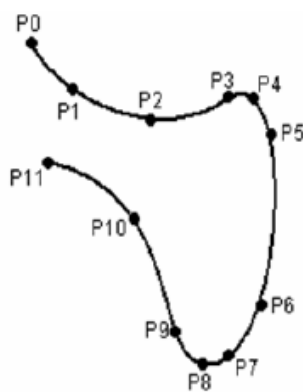
Spline es un tipo de movimiento especialmente apropiado para trayectorias curvas complejas. En principio, dichas trayectorias también se pueden crear con movimientos LIN y CIRC aproximados; no obstante, el Spline presenta una serie de ventajas.

El movimiento Spline es más versátil es el bloque Spline. Con el bloque Spline, se agrupan varios movimientos en un mismo movimiento general. La unidad de control de robot configura y ejecuta el bloque Spline como 1 conjunto de movimientos.

Los movimientos que se pueden incluir en un bloque Spline se denominan segmentos Spline. Estos se programan uno después de otro por aprendizaje. Un bloque Spline CP puede incluir segmentos SPL, SLIN y SCIRC. Un bloque Spline PTP puede incluir segmentos SPTP.

Aparte de los bloques Spline, también se pueden programar movimientos individuales Spline: SLIN, SCIRC y SPTP.

5.3.5.1. Ventajas del bloque Spline



La trayectoria se define con puntos que se encuentran en la propia trayectoria. La trayectoria deseada se crea fácilmente.

La velocidad programada se mantiene mejor que en los tipos de movimientos convencionales. Solo en muy pocos casos se produce una reducción de la velocidad. En los bloques Spline CP, además se pueden definir zonas especiales de desplazamiento constante.

El curso de la trayectoria siempre es la misma, independientemente del override, de la velocidad o de la aceleración.

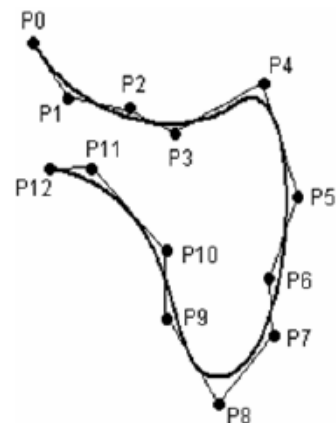
Ilustración 42 Ventajas Spline

Los círculos y radios estrechos se recorren con gran precisión.

5.3.5.2. Desventajas LIN/CIRC

La trayectoria se define con puntos aproximados que no se encuentran en la propia trayectoria. Las zonas de posicionamiento aproximado son difíciles de predecir. Resulta costoso crear la trayectoria deseada.

En muchos casos se producen reducciones de velocidad difíciles de predecir como por ejemplo en las zonas de posicionamiento aproximado y en caso de puntos muy próximos.



El curso de la trayectoria cambia si no se puede llevar a cabo la aproximación, p. ej., por motivos de tiempo. *Ilustración 43 Desventajas LIN/CIRC*

El curso de la trayectoria varía en función del override, la velocidad o la aceleración.

5.4. Programación lógica

Vamos a pasar a explicar las funciones básicas y la sintaxis básica KRL de la programación para poder crear un programa.

5.4.1. Grupos de usuario

En el KUKA Software System se dispone de distintas funciones dependiendo del grupo de usuario seleccionado. Existen los siguientes grupos usuarios:

- **Operario:** Grupo para el operario. Este es el grupo de usuario por defecto.
- **Usuario:** Grupo para el operario. (Los grupos de usuario del operario y del usuario están declarados por defecto para el mismo grupo destinatario).
- **Experto:** Grupo de usuario para el programador. Este grupo de usuario está protegido por una contraseña.
- **Técnico de mantenimiento de seguridad I:** Este grupo de usuario puede activar y configurar la configuración de seguridad del robot. Este grupo de usuario está protegido por una contraseña.
- **Técnico de mantenimiento de seguridad II:** Este grupo de usuario solo es relevante cuando se utiliza KUKA SafeOperation o KUKA SafeRangeMonitoring. Este grupo de usuario está protegido por una contraseña.
- **Administrador:** Mismas funciones a las del grupo de usuario “Experto”. A modo adicional es posible integrar plug-ins en la unidad de control del robot. Este grupo de usuario está protegido por una contraseña.

5.4.2. Estructura de un programa

En la unidad de control de robot están en curso a la vez dos tareas: el interpretador Submit y el interpretador de Robot.

A través del interpretador Submit se ejecuta un programa SUB. Un programa SUB puede realizar tareas de servicio y control como por ejemplo: control de dispositivos de protección, control de un circuito de refrigeración, etc. El interpretador Submit se activa de forma automática al conectar la unidad de control del robot. Para ello se inicia el programa SPS.SUB. Los programas SUB son siempre ficheros con extensión *.SUB. Es posible editar el programa SPS.SUB y pueden crearse otros programas SUB.

En un programa SUB pueden utilizarse casi todas las instrucciones KRL. No obstante, un programa SUB no puede contener instrucciones que hacen referencia a movimientos de robot, ni puede activar ningún subprograma que contenga instrucciones de movimiento.

A través del interpretador del robot se ejecuta el programa de movimiento. Estos programas KRL están formados por un fichero *.SRC y otro *.DAT con el mismo nombre.

El archivo DAT contiene los datos permanentes y coordenadas de puntos, mientras que el fichero SRC contiene el código del programa que cuenta con la siguiente estructura (el archivo DAT no es necesario para la correcta ejecución del programa, de hecho existe el archivo \$CONFIG.DAT en el que podemos almacenar todos los datos de manera global):

```
1 DEF my_program( )
2  INI
3
4  PTP HOME Vel= 100 % DEFAULT
5  ...
6  LIN point_5 CONT Vel= 2 m/s CPDAT1 Tool[3] Base[4]
7  ...
14 PTP point_1 CONT Vel= 100 % PDAT1 Tool[3] Base[4]
15 ...
20 PTP HOME Vel= 100 % DEFAULT
21
22 END
```

Ilustración 44 Estructura programa .SRC

Cuenta básicamente con un encabezado con el nombre del programa, una sección INI que contiene inicializaciones para variables internas y parámetros, las instrucciones lógicas y de movimiento, y la finalización que es la línea END

La primera instrucción de movimiento de un programa KRL debe ser una posición de salida inequívoca. La posición HOME, creada por defecto en la unidad de control del robot, garantiza este precepto. Si la primera instrucción no es la posición HOME o esta ha sido cambiada, debe utilizar una instrucción de movimiento PTP.

La llamada a un subprograma se realiza escribiendo el nombre del programa seguido de unos paréntesis. Una vez que el subprograma se ha procesado, se vuelve a la línea siguiente de programa en la que se realizó la llamada.

5.4.3. Tipos de datos

En el lenguaje KRL existen los siguientes tipos de datos:

- Tipos de datos definidos por el usuario: se derivan de tipos de datos STRUCT o ENUM
- Tipos de datos simples: INT, REAL, BOOL y CHAR
- Tipos de datos para la programación de movimiento: AXIS, E6AXIS, FRAME, POS y E6POS

5.4.4. Declaración de variables

Las variables en KRL se declaran de la siguiente manera:

DECL + “Tipo de Variable” + “Nombre”

El nombre no debe tener más de 24 caracteres, no deben empezar por un número ni por el símbolo \$ (reservado para las variables del sistema).

Una vez definida una variable debemos asignarle un valor, en caso contrario su posición en memoria se define como no válido y no se nos guardará el valor al terminar el programa. Los tipos de variables más usados son INT, REAL, BOOL, CHAR, STRUCT y SIGNAL.

5.4.5. Operadores

En cada operación, el compilador comprueba la admisibilidad de los operandos.

5.4.5.1. Operadores aritméticos

En KRL están permitidos los cuatro tipos de cálculos básicos.

Operador	Descripción
+	Suma o signo positivo
-	Resta o signo negativo
*	Multiplicación
/	División

Tabla 10 Operadores aritméticos

Los operadores aritméticos se pueden aplicar a los tipos de datos INT y REAL. Si el resultado de una división INT no es un número entero, se eliminarán las posiciones decimales.

Operando	Operando	Resultado
INT	INT	INT
INT	REAL	REAL
REAL	REAL	REAL

Tabla 11 Resultados operadores aritméticos

5.4.5.2. Operador geométrico

Con los operadores geométricos se pueden sumar posiciones geométricas. La suma geométrica también se denomina “combinación de Frame”. El operador geométrico se simboliza en KRL con el símbolo de dos puntos “:”.

El operador geométrico puede combinar los tipos de datos FRAME y POS/E6POS. El resultado siempre tiene el mismo tipo de dato que el operando que, en el lado derecho, más alejado se encuentre.

Combinación de dos operandos:

A la izquierda	:	A la derecha	Resultado
POS	:	POS	POS
POS	:	FRAME	FRAME
FRAME	:	FRAME	FRAME
FRAME	:	POS	POS

Tabla 12 Combinación operadores geométricos

Ejemplos para la combinación de tres operandos:

A la izquierda	:	Centro	:	A la derecha	Resultado
POS	:	POS	:	POS	POS
POS	:	POS	:	FRAME	FRAME
POS	:	FRAME	:	FRAME	FRAME
FRAME	:	FRAME	:	POS	POS

Tabla 13 combinación operadores geométricos

5.4.5.3. Operadores de comparación

Con los operadores de comparación se pueden formar expresiones lógicas. El resultado de una comparación siempre es del tipo BOOL.

Operador	Descripción	Tipos de datos admisibles
==	igual	INT, REAL, CHAR, ENUM, BOOL
<>	desigual	
>	mayor que	
<	menor que	
>=	mayor o igual que	
<=	menor o igual que	

Tabla 14 Operadores de comparación

5.4.5.4. Operadores lógicos

Los operadores lógicos sirven para la combinación lógica de las variables booleanas, constantes y expresiones lógicas simples, tal como se forman con la ayuda de los operadores de comparación.

Operador	Número de operandos	Descripción
NOT	1	Inversión
AND	2	Y lógico
OR	2	O lógico
EXOR	2	O exclusivo

Tabla 15 Operadores lógicos

Los operandos de una combinación lógica deben ser del tipo BOOL. El resultado también es siempre del tipo BOOL. La siguiente tabla muestra los resultados de las posibles combinaciones:

Operación		NOT A	A AND B	A OR B	A EXOR B
A = TRUE	B = TRUE	FALSE	TRUE	TRUE	FALSE
A = TRUE	B = FALSE	FALSE	FALSE	TRUE	TRUE
A = FALSE	B = TRUE	TRUE	FALSE	TRUE	TRUE
A = FALSE	B = FALSE	TRUE	FALSE	FALSE	FALSE

Tabla 16 Combinaciones operadores lógicos

5.4.6. Control de ejecución del programa

Se definirán las instrucciones más relevantes para controlar la ejecución del programa del robot de paletizado.

5.4.6.1. *CONTINUE*

Con CONTINUE se puede evitar una parada del movimiento de avance que podría producirse en la siguiente línea de programa. CONTINUE siempre hace referencia a la siguiente línea, incluso si se trata de una línea vacía.

```
CONTINUE
$OUT[1]=TRUE
CONTINUE
$OUT[2]=FALSE
```

En este caso, las salidas se colocan en la ejecución en avance. El momento preciso en el que se ajustan no se puede prever.

5.4.6.2. *FOR ... TO ... ENDFOR*

Se ejecutará un bloque de instrucciones tan a menudo hasta que un contador sobrepase por encima o por debajo un valor definido

Después de la última ejecución del bloque de instrucciones, el programa continuará procesándose con la primera instrucción tras el ENDFOR. Es posible abandonar el bucle antes de tiempo con EXIT. Los bucles se pueden intercalar. En caso de bucles intercalados, primero se recorre en bucle externo completo. A continuación se recorre el bucle interno completo.

La sintaxis queda de la siguiente manera:

```
FOR Contador = Valor inicial TO Valor final <STEP Amplitud del paso>
<Instrucciones>
ENDFOR
```

Contador: es una variable del tipo INT que cuenta las ejecuciones. Se predefine con el Valor inicial. Antes deberá declararse la variable. El valor de contador puede emplearse en instrucciones dentro y fuera del bucle. Después de salir del bucle, el contador adopta el último valor tomado.

El contador debe predefinirse con el Valor inicial. Después de cada ejecución del bucle el contador cambia automáticamente en la amplitud de paso. En el momento en que se supera por encima o por debajo el Valor final se da por finalizado el bucle.

Amplitud de paso es un valor entero por el que el contador cambiará en cada ejecución del bucle. El valor puede ser negativo, pero no puede ser cero ni una variable.

```
INT A
...
FOR A=1 TO 10 STEP 2
  B=B+1
ENDFOR
```

La variable B se aumentará en 1 cada 2 ejecuciones.

5.4.6.3. *HALT*

Detiene el programa. La última instrucción de movimiento lanzada se ejecuta por completo. El programa puede continuarse exclusivamente con la tecla de arranque. A continuación se ejecutará la siguiente instrucción después de HALT.

En un programa de interrupción el programa no se detiene hasta que se haya ejecutado por completo el procesamiento.

5.4.6.4. *IF ... THEN ... ENDIF*

Ramificación condicionada. En función de una condición se ejecuta o el primero bloque de instrucciones (bloque THEN) o el segundo bloque de instrucciones (bloque ELSE). Luego el programa se reanuda tras ENDIF.

El bloque ELSE puede faltar. En caso de una condición que no se cumple, el programa puede proseguir de inmediato después de ENDIF.

El número de instrucciones de los bloques es ilimitado. Se pueden intercalar varias instrucciones IF entre sí.

```
IF Condición THEN
Instrucciones
<ELSE
Instrucciones>
ENDIF
```

La condición puede ser una variable de tipo BOOL, una función de tipo BOOL o una combinación, por ejemplo una comparación, con resultado de tipo BOOL.

```
IF $IN[1]==TRUE THEN
  $OUT[17]=TRUE
ELSE
  $OUT[17]=FALSE
ENDIF
```

5.4.6.5. *SWITCH ... CASE ... ENDSWITCH*

Selecciona uno de varios posibles bloques de instrucciones, siguiendo un criterio de selección. Cada bloque de instrucciones tiene al menos una identificación. Se elegirá el bloque cuya identificación coincida con el criterio de selección.

Si el bloque se ejecuta, la ejecución del programa continuará tras ENDSWITCH.

Si no coincide ninguna identificación con el criterio de selección, se ejecutará el bloque DEFAULT. Si no hay ningún bloque DEFAULT disponible, no se ejecutará ningún bloque y se continuará con la ejecución del programa tras ENDSWITCH.

```

SWITCH Criterio de selección
CASE Identificación1 <, Identificación2, ... >
Bloque de instrucciones
<CASE IdentificaciónM <, IdentificaciónN, ... >
Bloque de instrucciones >
<DEFAULT
Bloque de instrucciones por defecto>
ENDSWITCH

```

Entre la línea SWITCH y la primera línea CASE no puede haber ni una línea vacía ni comentario. Dentro de una instrucción SWITCH sólo puede aparecer una vez DEFAULT.

El criterio de selección puede ser una variable, una llamada de función o una expresión del tipo de datos INT, CHAR o ENUM. El tipo de datos de la identificación debe coincidir con el del criterio de selección. Un bloque de instrucciones puede tener todas las identificaciones que desee. Si hay varias identificaciones deberán separarse entre sí mediante una coma.

```

INT VERSION
...
SWITCH VERSION
CASE 1
    UP_1()
CASE 2,3
    UP_2()
    UP_3()
    UP_3A()
DEFAULT
    ERROR_UP()
ENDSWITCH

```

5.4.6.6. WAIT FOR

Detiene el programa hasta que se cumple una determinada condición. Después, el programa prosigue.

```

WAIT FOR Condición

```

La condición con la que debe proseguir la ejecución del programa debe ser del tipo BOOL. Si al acceder a WAIT la condición ya es TRUE la ejecución del programa no se detiene. Sin embargo, si la condición es FALSE, la ejecución del programa se detiene hasta que la condición sea TRUE.

5.4.6.7. TRIGGER WHEN DISTANCE

El Trigger lanza una instrucción definida por el usuario. La unidad de control del robot ejecuta la instrucción de forma paralela al movimiento del robot. El Trigger puede referirse opcionalmente al punto de inicio o de destino del movimiento. La instrucción se puede activar directamente en el punto de referencia o aún se puede desplazar temporalmente.

```

TRIGGER WHEN DISTANCE=Posición DELAY=Tiempo DO Instrucción
<PRIO=Prioridad>

```

Si Posición es igual a 0 se referencia al punto de inicio, y si es igual a 1 se referencia al punto de destino.

Tiempo es un valor REAL que puede ser una variable, una constante o una función. Se define en milisegundos y es el desplazamiento en el tiempo con respecto a la posición. Si toma un valor negativo el desplazamiento es en dirección al principio del movimiento, mientras que si el valor es positivo el desplazamiento es en dirección al final del movimiento. Si no se desea un desplazamiento, ajustar Tiempo = 0.

La prioridad del Trigger solo es pertinente cuando la Instrucción llama a un subprograma. Si varios Trigger llaman a la vez a subprogramas, se procesará primero el Trigger con la prioridad más alta y luego el de menor prioridad. 1 = prioridad más alta.

El punto de conexión no puede desplazarse libremente en el espacio. La siguiente tabla indica los desplazamientos que son posibles como máximo. Si se programan desplazamientos mayores y, de este modo, no válidos, la unidad de control de robot conecta el Trigger muy tarde en el límite permitido.

DISTANCE = ...	Desplazamiento negativo máximo hasta ...	Desplazamiento positivo máximo hasta ...
DISTANCE = 0	<p>---</p> <p>(No es posible ningún desplazamiento negativo.)</p>	<ul style="list-style-type: none"> ■ Hasta el punto de destino ■ Si el punto de destino se ha programado por aproximación: Hasta el principio del arco de aproximación
DISTANCE = 1 y Punto de destino = parada exacta	<ul style="list-style-type: none"> ■ Hasta el punto de inicio ■ Si el punto de inicio se ha programado por aproximación: Hasta el final del arco de aproximación 	<p>---</p> <p>(No es posible ningún desplazamiento positivo.)</p>
DISTANCE = 1 y Punto de destino = programado por aproximación	Hasta el principio del arco de aproximación del punto de destino	Hasta el final del arco de aproximación del punto de destino

Tabla 17 Desplazamientos Trigger

130 milisegundos tras P_2 se pone \$OUT[8] en TRUE

```

LIN P_2
TRIGGER WHEN DISTANCE=0 DELAY=130 DO $OUT[8]=TRUE
LIN P_3

```

En la mitad del arco de aproximación de P_5 se activa el subprograma MY_SUBPROG con la prioridad 5.

```

PTP P_4
TRIGGER WHEN DISTANCE=1 DELAY=0 DO MY_SUBPROG() PRIO=5
PTP P_5 C_DIS
PTP P_6

```

5.4.6.8. *INTERRUPT ... DECL ... WHEN ... DO*

Si se da un suceso determinado, por ejemplo una entrada, la unidad de control interrumpe el programa actual y ejecuta un subprograma predefinido. El suceso y el subprograma se definen con `INTERRUPT ... DECL ... WHEN ... DO`.

Cuando se ha ejecutado el subprograma se continuará el programa interrumpido desde el momento de la interrupción. Excepción: `RESUME`.

Se pueden declarar como máximo 32 interrupciones simultáneamente. En todo momento es posible sobrescribir una declaración de interrupción por otra nueva.

```
<GLOBAL> INTERRUPT DECL Prio WHEN Suceso DO Subprograma
```

Una interrupción se reconoce a partir del nivel en el cual fue declarada. Esto significa: una interrupción declarada en un subprograma, no es conocida en el programa principal (y tampoco puede activarse ahí). Si una interrupción tiene que conocerse también en niveles superiores, entonces debe anteponerse a la declaración la palabra `GLOBAL`.

Si se producen simultáneamente varias interrupciones, se procesará en primer lugar la interrupción que tenga la prioridad más alta y a continuación las de prioridades más bajas. 1 = prioridad más alta.

El suceso con el que debe darse la interrupción debe ser del tipo `BOOL`. Son admisibles una variable global booleana, un nombre de señal, una comparación. Una combinación lógica simple.

Declaración de una interrupción con la prioridad 23, que llama al subprograma `UP1`, si `$IN[12]` es verdadero. Se transmiten al subprograma los parámetros 20 y `VALUE`.

```
INTERRUPT DECL 23 WHEN $IN[12]==TRUE DO UP1(20,VALUE)
```

Las interrupciones deben activarse y desactivarse por medio de la siguiente instrucción:

```
INTERRUPT Acción <Número>
```

En el término *Acción* debemos elegir `ON`, `OFF` y en *Número* la prioridad de la interrupción declarada.

La instrucción `RESUME` solo puede estar presente en programas de interrupción. `RESUME` cancela todos los programas de interrupción y subprogramas en curso hasta el nivel en el cual se encuentra declarada la interrupción actual.

6. Descripción del software del robot

En este apartado vamos a proceder a explicar el desarrollo del programa que va a dar vida al brazo robotizado de la instalación.

Primero vamos a ver la configuración del modo automático externo, la configuración de los mosaicos y las señales que intercambia con la red.

Posteriormente explicaremos cada uno de los programas y subprogramas que conforman el proyecto del robot.

6.1. Configuración Automático Externo

Lo que vamos a tratar en este apartado son las configuraciones necesarias para que el controlador **KRC4** pueda ser gobernado mediante el PLC y el intercambio de señales que se necesitan para tal fin.

El sistema KUKA cuenta con un programa llamado “**CELL.SRC**” destinado a ejecutar distintos programas mediante órdenes del PLC. Nosotros para nuestro proyecto no hemos utilizado esta opción que nos da el sistema y hemos seleccionado nuestro programa “**principal**” de modo que desde el PLC siempre damos marcha a este programa.

Para configurar las señales de automático externo debemos irnos desde el menú a **Configurar > Entradas/Salidas > Automático Externo** y se nos mostrará la ventana de configuración:

	Denominación	Tipo	Nombre	Valor
1	Tipo de nº de programa	Var	PGNO_TYPE	1
2	Número de programa reflejado	Var	REFLECT_PROG_I	0
3	Ancho de bit del nº de programa	Var	PGNO_LENGTH	8
4	Primer bit de nº de programa	BO	PGNO_FBIT	33
5	Bit de paridad	BO	PGNO_PARITY	41
6	Nº de programa válido.	BO	PGNO_VALID	42
7	Inicio de programa	BO	SEXT_START	1026
8	Movimiento habilitado	BO	\$MOVE_ENABLE	1025
9	Confirmación de error	BO	\$CONF_MESS	1026
10	Accionamientos desconectados (invertido)	BO	\$DRIVES_OFF	1025
11	Accionamientos ON	BO	\$DRIVES_ON	140
12	Activar interfaz	BO	\$I_O_ACT	1025

Ilustración 45 configuración automático externo

6.1.1. Entradas

\$IN	DESCRIPCIÓN
1	\$EXT_START : Pulso marcha Automático Externo del programa del Robot
2	\$MOVE_ENABLE : Habilitación de movimiento del Robot
3	\$CONF_MESS : Pulso borrado mensajes de fallo del Robot
4	\$DRIVES_OFF : Habilitar los accionamientos del Robot
5	\$DRIVES_ON : Pulso conectar los accionamientos del Robot
6	\$I_O_ACT : Activación de la interface de Automático Externo
21	Orden Marcha Robot en Modo Automático Externo

Tabla 18 Entradas Automático Externo

6.1.2. Salidas

\$OUT	DESCRIPCIÓN
1	\$T1 : Modo de Servicio Test 1
2	\$T2 : Modo de Servicio Test 2
3	\$AUT : Modo de Operación Automático
4	\$EXT : Modo de Operación Automático Externo
5	\$RC_RDY1 : Control preparado
6	\$ALARM_STOP : Circuito de parada de emergencia cerrado
7	\$USER_SAF : Protección del operario cerrada
8	\$PERI_RDY : Accionamientos preparados
9	\$ROB_CAL : Ejes del Robot ajustados
10	\$I_O_ACTCONF : Interface Modo Automático Externo activada
11	\$STOPMESS : Avería general (mensaje que requiere la parada del Robot)
12	\$ALARM_STOP_INTERN : Seta Smart PAD no accionada
13	\$PRO_ACT : Proceso activo
14	\$APPL_RUN : Programa de aplicación en marcha
15	\$PRO_MOVE : Robot en movimiento
16	\$IN_HOME : Robot en posición inicial (HOME)
22	\$ON_PATH : Posición del Robot dentro de la trayectoria programada
23	\$NEAR_POSRET : Posición del Robot cerca de la trayectoria programada
24	\$ROB_STOPPED : Robot detenido (no en movimiento)

Tabla 19 Salidas Automático Externo

6.1.3. Secuencia de inicio

El siguiente gráfico muestra la secuencia que debe seguirse para iniciar un programa en Automático Externo desde el PLC.

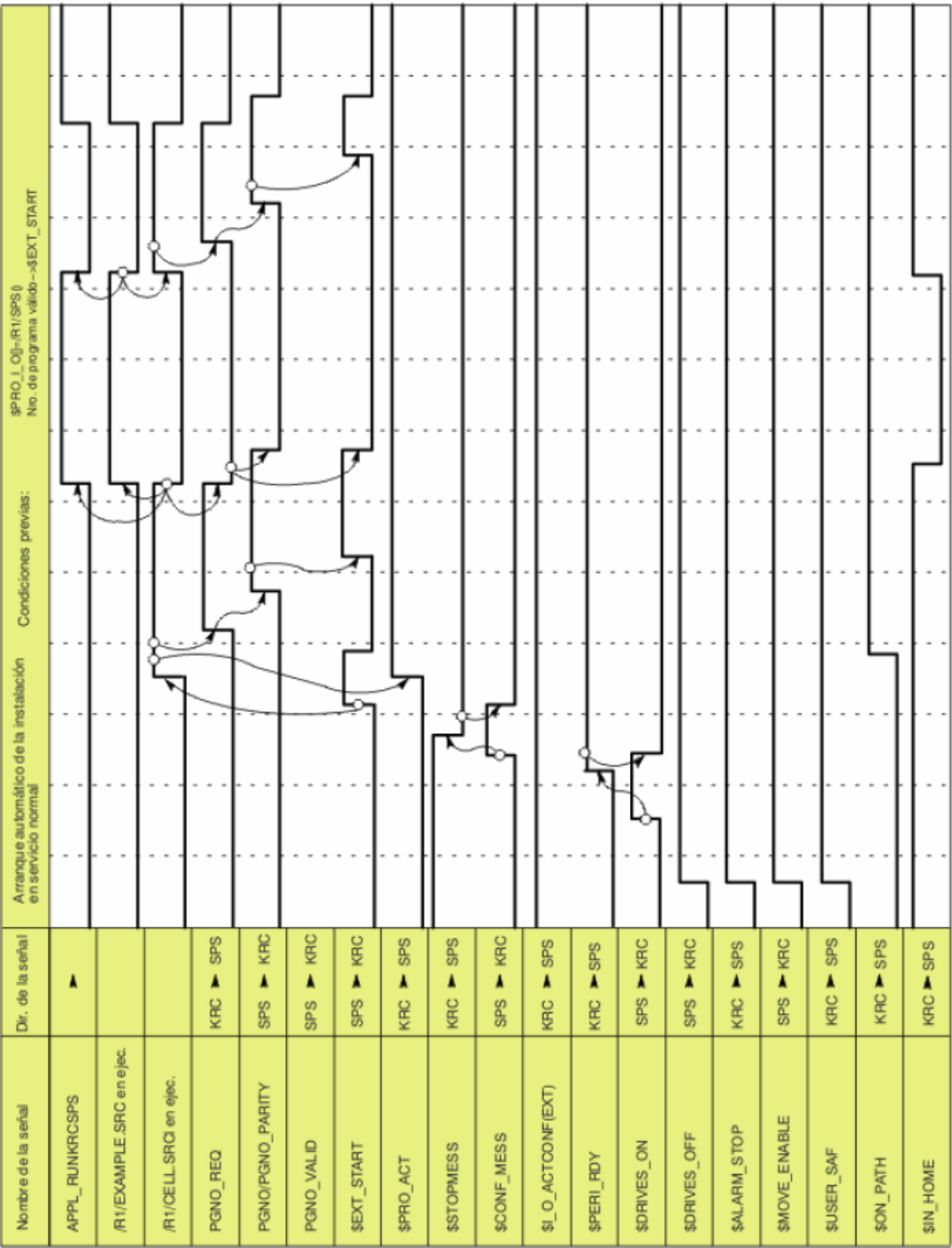


Ilustración 46 Secuencia inicio Automático Externo

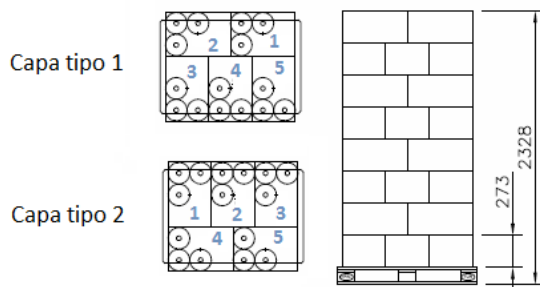
6.2. Configuración de los mosaicos

Vamos a mostrar como quedarían los mosaicos de los distintos productos de la línea 1 y 2 en el europalet y el palet doble.

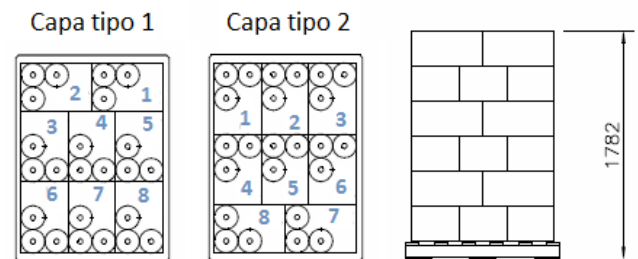
6.2.1. Productos de la línea 1

Papel Higiénico Industrial 320 g 18 rollos.

Europalet:

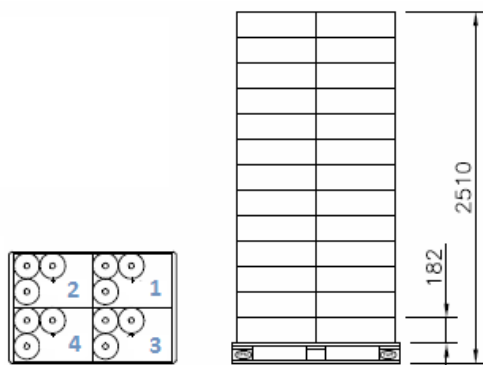


Palet doble:

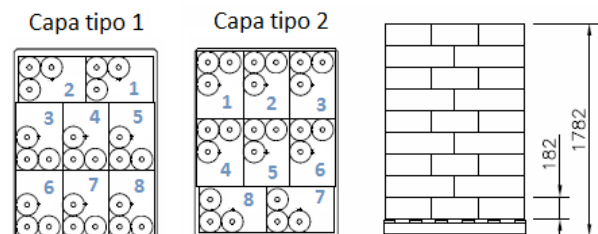


Papel Higiénico Industrial 400 g 12 rollos.

Europalet:

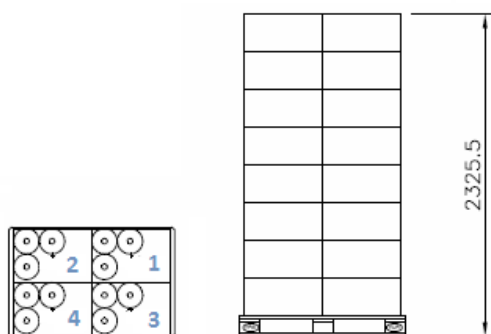


Palet doble:

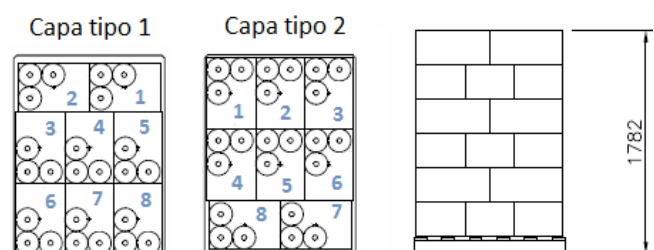


Papel Higiénico Industrial 400 g 18 rollos.

Europalet:

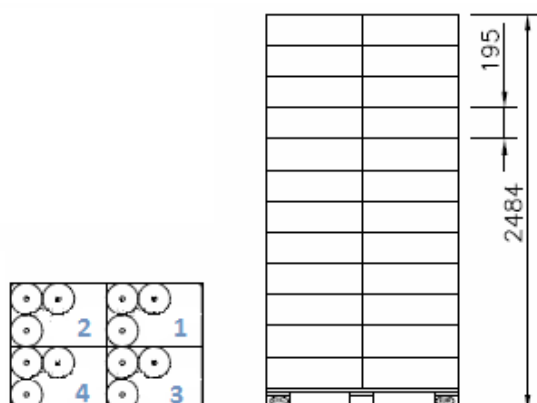


Palet doble:



Secamanos 1kg

Europalet:

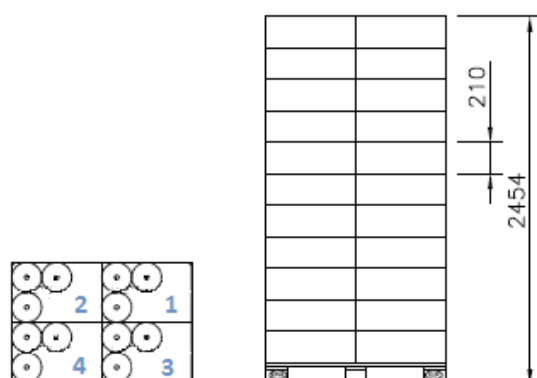


Palet doble:

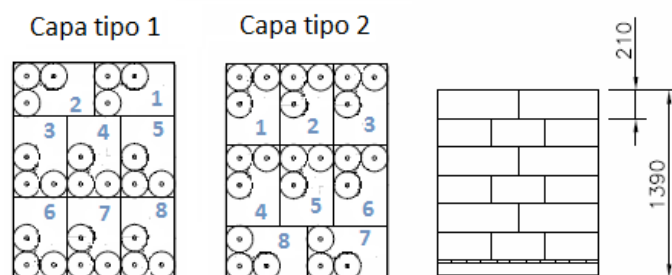


Secamanos 1.2kg

Europalet:

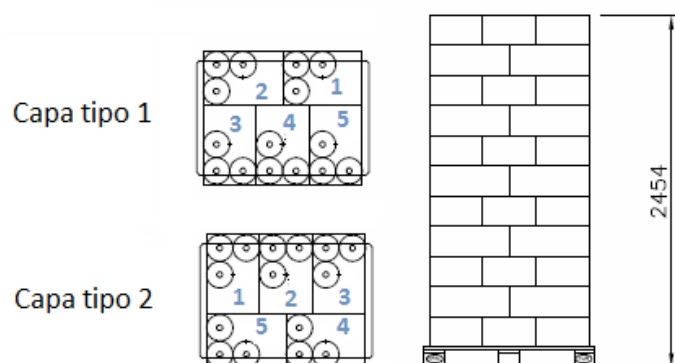


Palet doble:



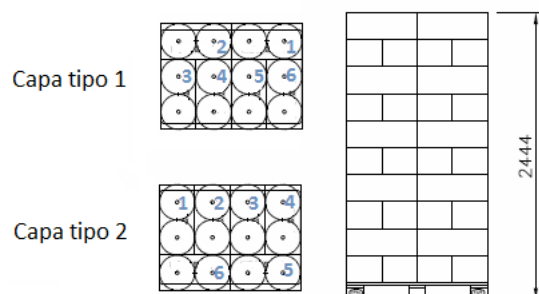
Maxi-roll 600 gr

Solo se paletiza en europalet:



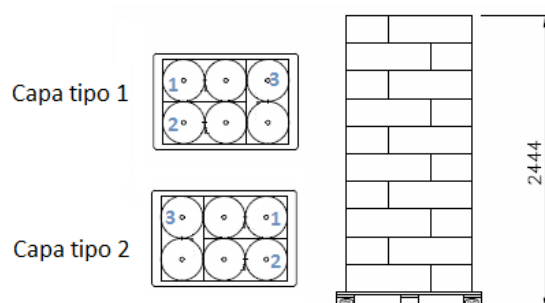
Papel mecánico Ø300

Solo se paletiza en europalet:



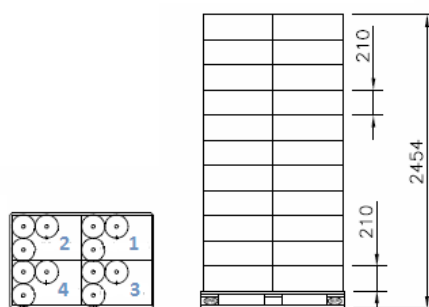
Papel mecánico Ø350

Solo se paletiza en europalet:



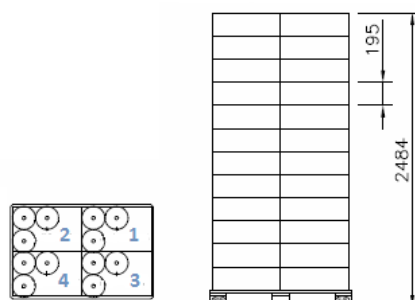
Multisusos individual 6 rollos.

Solo se paletiza en europalet:



Secamanos 850 gr

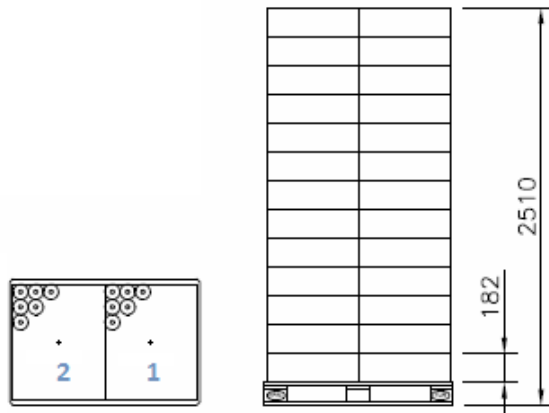
Solo se paletiza en europalet:



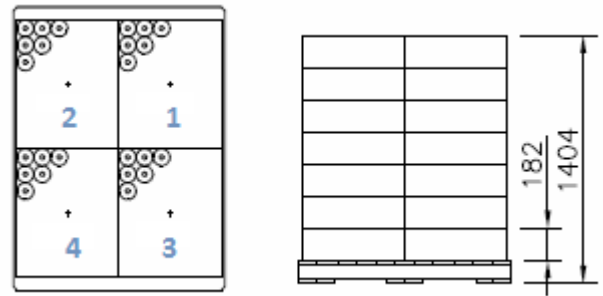
6.2.2. Productos de la línea 2

Papel Higiénico 15 metros

Europalet:

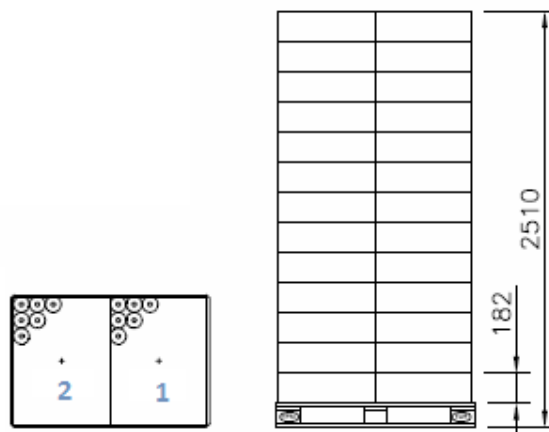


Palet doble:

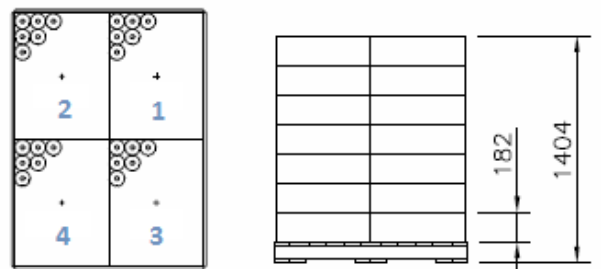


Papel Higiénico 16 metros

Europalet:

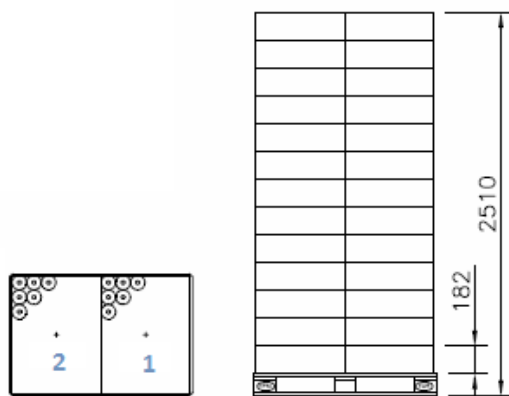


Palet doble:

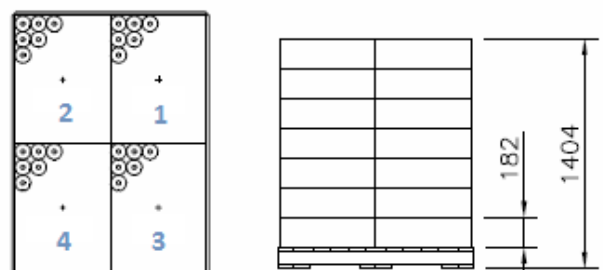


Papel Higiénico 18 metros

Europalet:

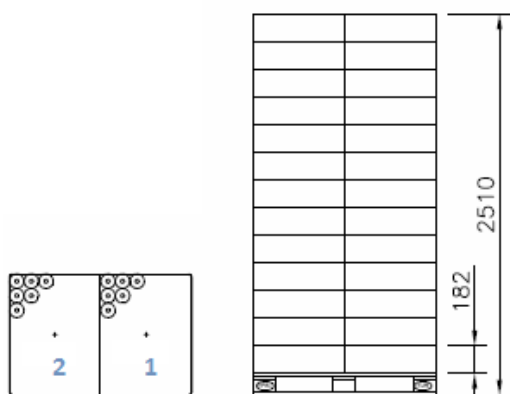


Palet doble:

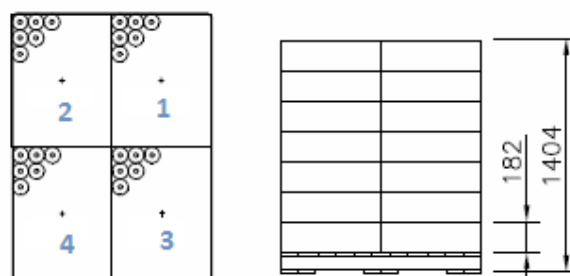


Papel Higiénico 22 metros:

Europalet:

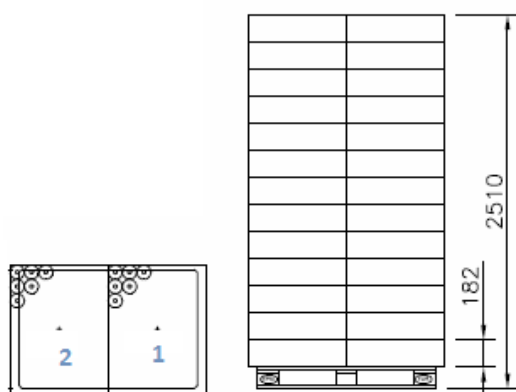


Palet doble:

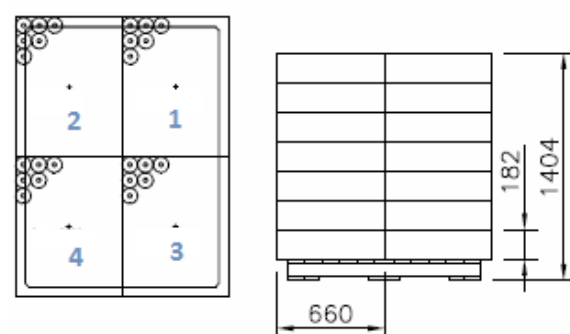


Papel Higiénico 35 metros:

Europalet:



Palet doble:



6.3. Lista de señales con el PLC

6.3.1. Entradas

§IN	NOMBRE	DESCRIPCION
21	PERMISO_MOV_ROB	Orden Marcha Robot en Modo Automático Externo
25	PINZA_ABIERTA	Pinza paletizado abierta sin incoherencia
26	PAQUETE_COGIDO	Paquete cargado sin incoherencia (pinza cerrada)
27	GARRA_ABIERTA	Garra de palets abierta sin incoherencia
28	GARRA_CERRADA	Garra de palets totalmente cerrada sin incoherencia
29	PALET_COGIDO	Palet cargado en las garras sin incoherencia
30	PALPADOR_EXTENDIDO	Palpador de palets extendido sin incoherencia
31	PALPADOR_RECOGIDO	Palpador de palets recogido sin incoherencia
32	PALET_DETECTADO	Palet palpado
33	PERMISO_CARGAR_PAQUETE	Permiso cargar paquete de Línea 1 ó 2
34	COGIDA_LINEA1_GIRADA	Cargar Paquete de Línea 1 con Pinza girada
35	PERMISO_CARGAR_PALET	Permiso cargar palet
36	PERMISO_ENTRAR_PALET	Permiso entrada en zona depositado palet (hasta punto predepositado)
37	PERMISO_DEJAR_PALET	Permiso depositar palet sobre Mesa de Paletizado
38	PERMISO_DEJAR_PAQUETE	Permiso depositar paquete sobre el palet
40	RESET_POS_PALET	Orden iniciar medición altura torre palets
41	ROBOT_HOME	Orden llevar el Robot a posición HOME
42	TIPO_EUROPALET	Orden ir a cargar palet tipo 1 (europalet)
43	TIPO_DOBLEPALET	Orden ir a cargar palet tipo 2 (paleta grande)
44	CARGAR_LINEA1	Orden ir a cargar paquete de Línea 1
45	CARGAR_LINEA2	Orden ir a cargar paquete de Línea 2
47	ABORTAR_TAREA	Orden abortar operación en curso
48	PETICION_ACCESO	Petición de acceso al recinto vallado del Robot
55	CONF_PALET_DEPOSITADO	Enterado palet depositado sobre Mesa de Paletizado
56	CONF_PAQUETE_DEPOSITADO	Enterado paquete depositado sobre el palet
64	DATOS_OK	Los datos enviados al Robot son válidos
65-80	VEL_ROBOT	Velocidad general del Robot (%)
81-96	CAPA_DESTINO	No. Capa destino de depositado del producto en el palet
97-112	PAQUETE_DESTINO	No. Paquete destino de depositado del producto en el palet
113-128	ALT_PAQUETE	Altura del Paquete (mm)

129-144	X_PUNTO_COGER1_NORMA L	Carga Paquete L1 (Normal): Coordenada X (mm)
145-160	Y_PUNTO_COGER1_NORMA L	Carga Paquete L1 (Normal): Coordenada Y (mm)
161-176	X_PUNTO_COGER1_GIRADO	Carga Paquete L1 (Girada): Coordenada X (mm)
177-192	Y_PUNTO_COGER1_GIRADO	Carga Paquete L1 (Girada): Coordenada Y (mm)
193-208	OFFSET_Z_CARGA	Offset Z del punto de carga del paquete en Línea 1 ó 2 (mm)
209-224	X_PUNTO	Coordenada X punto depositado paquete (mm)
225-240	Y_PUNTO	Coordenada Y punto depositado paquete (mm)
241-256	Z_PUNTO	Coordenada Z punto depositado paquete (mm)
257-272	A_PUNTO	Coordenada A punto depositado paquete (º)
273-288	OFF_X	Offset X punto separación depositado paquete (mm)
289-304	OFF_Y	Offset Y punto separación depositado paquete (mm)

Tabla 20 Listado Entradas PLC

6.3.2. Salidas

\$OUT	NOMBRE	DESCRIPCION
25	ABRIR_PINZA	Orden abrir pinza paletizado
26	CERRAR_PINZA	Orden cerrar pinza paletizado
27	ABRIR_GARRA	Orden abrir garras toma de palet
28	CERRAR_GARRA_TOMAR_PALET	Orden cerrar garras para tomar palet
29	CERRAR_GARRA	Orden cerrar garras toma palet para esconder
30	EXTENDER_PALPADOR	Orden extender palpador palets
31	RECOGER_PALPADOR	Orden recoger palpador palets
32	ANULAR_PALAS_CORTAS	Orden anular accionamiento de palas cortas pinza paletizado
33	ENTERADO_PAQUETE_CARGADO O	Enterado pinzas con paquete cargado de Línea 1 ó 2
35	ENTERADO_PALET_CARGADO	Enterado garras con palet cargado
36	PALET_PRE_DEPOSITADO	En posición previa al depositado del palet
40	RESET_HECHO	Enterado orden iniciar medición altura torre palets
41	PROG_HOME	Enterado orden llevar Robot a posición HOME
42	PROG_EPALET	Enterado orden ir a cargar palet tipo 1 (europalet)
43	PROG_DPALET	Enterado orden ir a cargar palet tipo 2 (paleta grande)
44	PROG_LINEA1	Enterado orden ir a cargar paquete de Línea 1
45	PROG_LINEA2	Enterado Orden ir a cargar paquete de Línea 2
47	TAREA_ABORTADA	Enterado orden abortar operación en curso
48	PERMISO_ACCESO	Permiso de acceso al recinto vallado del Robot
49	ROBOT_LINEA1	Robot en zona de interferencia con Línea 1
50	ROBOT_LINEA2	Robot en zona de interferencia con Línea 2
51	ROBOT_ALIMENTADOR	Robot en zona de interferencia con Alimentador de Palets
52	ROBOT_MESA	Robot en zona de interferencia con Mesa de Paletizado
55	PALET_DEPOSITADO	Palet depositado sobre Mesa de Paletizado
56	PAQUETE_DEPOSITADO	Paquete depositado sobre el palet
57	PUNTO_CARGAR_PALET	Robot situado en el punto destino de carga de palet
58	PUNTO_CARGAR_PAQUETE	Robot situado en el punto destino de carga de paquete
59	PUNTO_DESCAR_PAQUETE	Robot situado en el punto destino de descarga de paquete
65-80	VEL_ACT_ROBOT	Velocidad actual del Robot (%)
81-96	ECO_CAPA_DESTINO	Eco No. Capa destino de depositado del producto en el palet
97-112	ECO_PAQUETE_DESTINO	Eco No. Paquete destino de depositado del producto en el palet

Tabla 21 Listado salidas PLC

6.4. Estructura del programa

El programa de robot cuenta con un programa principal y varios subprogramas que aglutinan las diferentes rutinas de movimiento del robot. Vamos a realizar un diagrama de flujo para entender los programas más importantes.

En el programa **SUB.SPS** se están actualizando continuamente las variables necesarias desde el PLC y el puntero de avance del mosaico con las coordenadas del punto en cuestión al que le manda el PLC para depositar el paquete.

Así el programa principal la primera vez que se inicia se eleva desde la posición en la que está a una cota segura **P_SEG_HOME** y desde ahí se desplaza a la posición **HOME**. En la posición **HOME** se mantiene a la espera de una orden que pueden ser: cargar de la línea 1, cargar de la línea 2, cargar un europalet, cargar un palet doble o ir a posición home.

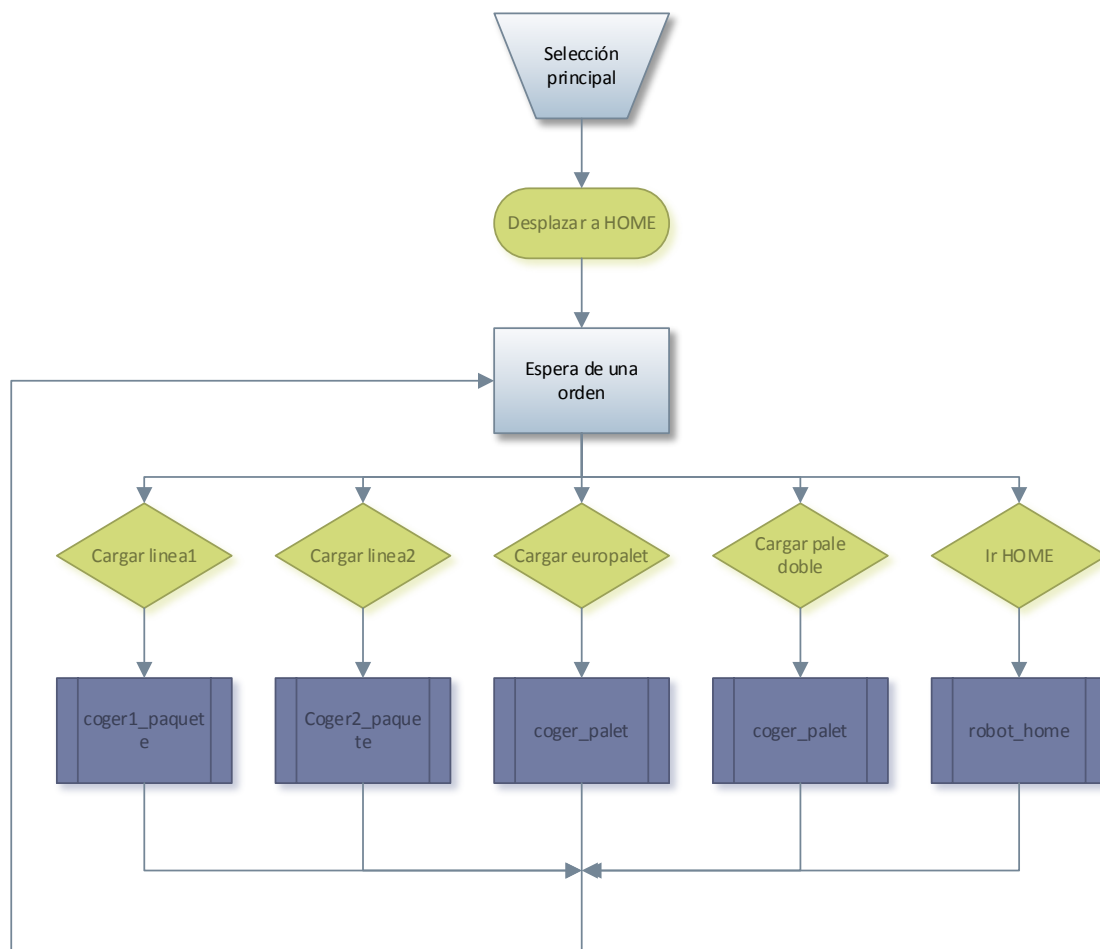


Ilustración 47 Diagrama flujo principal()

6.4.1. Programa coger línea 1 y coger línea 2

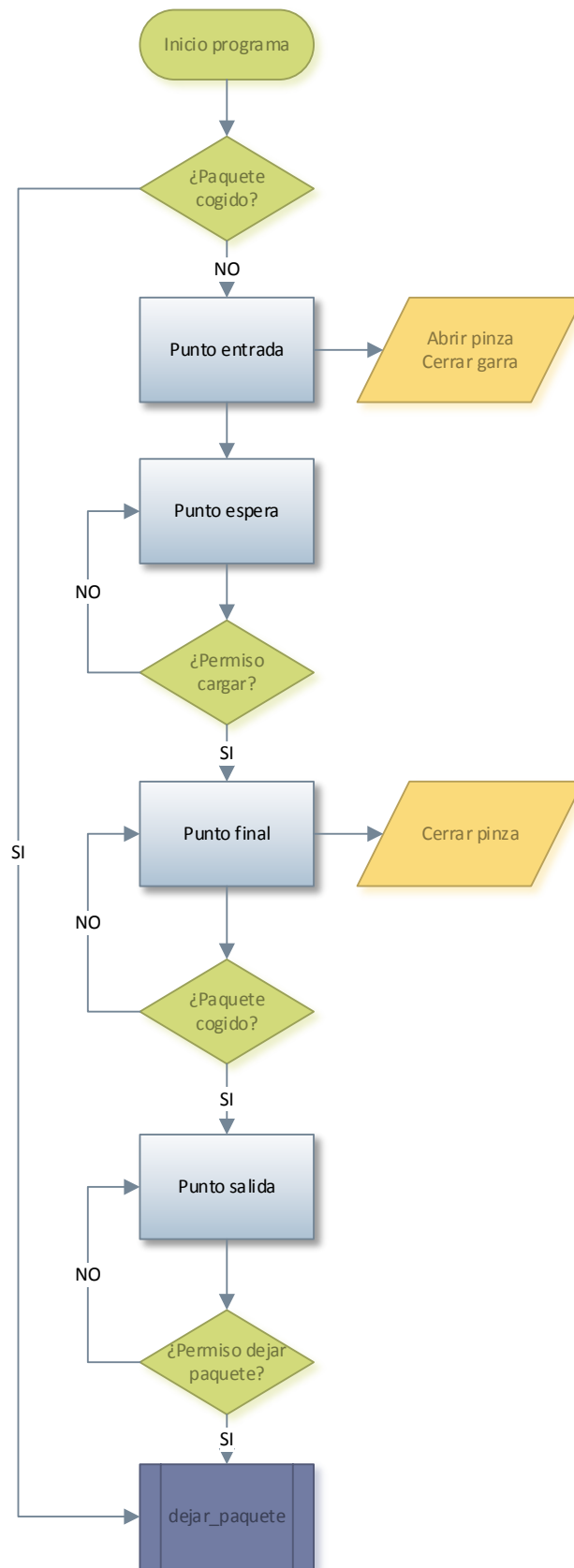


Ilustración 48 Diagrama flujo coger_paquete()

6.4.2. Programa coger europalet y palet doble

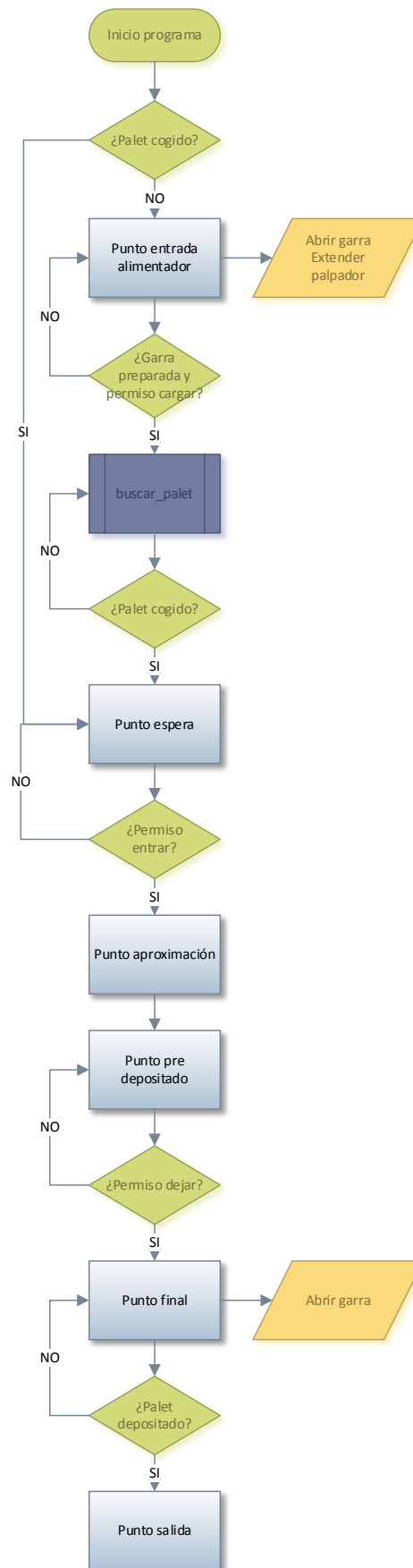


Ilustración 49 Diagrama flujo coger_palet()

6.4.3. Programa dejar paquete

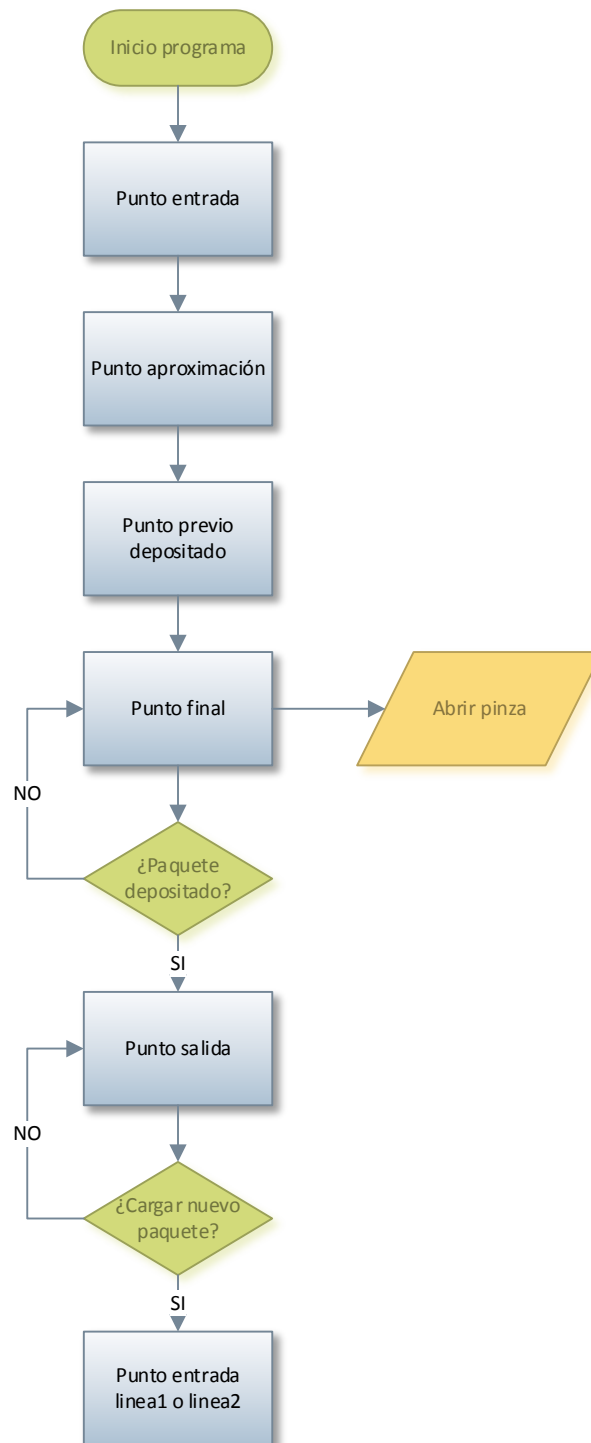


Ilustración 50 Diagrama flujo dejar_paquete

6.4.4. Programa buscar palet

Dentro de este programa se espera que se dé la interrupción **palet_detectado** que se activa cuando perdemos la señal del palpador. El programa de interrupción detiene el robot, recoge el palpador y guarda la posición actual para bajar más rápido la siguiente vez que busque un palet.

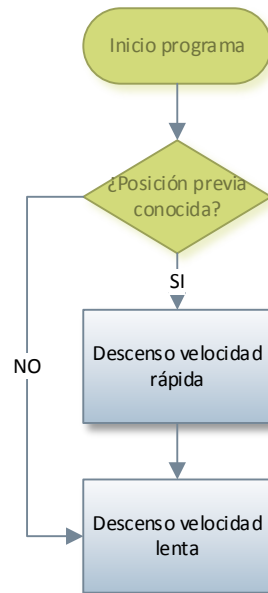


Ilustración 51 Diagrama flujo buscar_palet()

6.4.5. Programas de interrupción

En todo el transcurso del programa se contemplan varias interrupciones que pueden ocurrir en cualquier momento. Estas interrupciones son:

- **Abortar tarea** que detiene el movimiento del robot y lo lleva a una cota segura esperando y volviendo al programa principal esperando una nueva orden.
- **STOP_MOV** detiene el movimiento del robot y para el programa hasta que el operario lo inicie de nuevo.

7. Presupuesto

En este apartado detallaremos la estimación de los costes del proyecto en las siguientes tablas:

Costos Compras			
Máquina	Uds.	Valor unitario	Total
Robot de paletizado + garra + peana	1	240,00 €	240,00 €
Mesa dispensadora de torres palets	1	120,00 €	120,00 €
Mesa de paletizado	1	120,00 €	120,00 €
Mesa de salida almacén	1	120,00 €	120,00 €
Vallado para zona de paletizado	1	72,00 €	72,00 €
Transportes de producto	1	168,00 €	168,00 €
Subtotal			840,00 €

Tabla 22 Costes compras

Costos material			
Material	Uds.	Valor unitario	Total
Robot de paletizado + garra + peana	1	36.505,30 €	36.505,30 €
Mesa dispensadora de torres palets	1	1.943,28 €	1.943,28 €
Mesa de paletizado	1	1.295,00 €	1.295,00 €
Mesa de salida almacén	1	3.990,10 €	3.990,10 €
Vallado para zona de paletizado	1	2.030,00 €	2.030,00 €
Transportes de producto	1	12.926,83 €	12.926,83 €
Material eléctrico	1	13.058,98 €	13.058,98 €
Subtotal			71.749,49 €

Tabla 23 Costes material

Costos Mano Obra mecánica			
Máquina	Uds.	Valor unitario	Total
Robot de paletizado + garra + peana	1	3.756,30 €	3.756,30 €
Mesa dispensadora de torres palets	1	5.124,00 €	5.124,00 €
Mesa de paletizado	1	3.021,60 €	3.021,60 €
Mesa de salida almacén	1	7.752,00 €	7.752,00 €
Vallado para zona de paletizado	1	1.176,00 €	1.176,00 €
Transportes de producto	1	6.310,80 €	6.310,80 €
Subtotal			27.140,70 €

Tabla 24 Costes mano de obra mecánica

Costos Mano Obra eléctrica			
Máquina	Uds.	Valor unitario	Total
Robot de paletizado + garra + peana	1	3.360,00 €	3.360,00 €
Mesa dispensadora de torres palets	1	1.344,00 €	1.344,00 €
Mesa de paletizado	1	1.224,00 €	1.224,00 €
Mesa de salida almacén	1	1.272,00 €	1.272,00 €
Vallado para zona de paletizado	1	552,00 €	552,00 €
Transportes de producto	1	1.320,00 €	1.320,00 €
Subtotal			9.072,00 €

Tabla 25 Costes mano de obra eléctrica

Costos Ingeniería (Diseño planos eléctricos, mecánicos y automatización)			
Máquina	Uds.	Valor unitario	Total
Robot de paletizado + garra + peana	1	10.230,00 €	10.230,00 €
Mesa dispensadora de torres palets	1	4.521,00 €	4.521,00 €
Mesa de paletizado	1	4.174,50 €	4.174,50 €
Mesa de salida almacén	1	3.861,00 €	3.861,00 €
Vallado para zona de paletizado	1	1.848,00 €	1.848,00 €
Transportes de producto	1	2.211,00 €	2.211,00 €
Subtotal			26.845,50€

Tabla 26 Costes Ingeniería planos eléctricos, mecánicos y automatización

COSTOS FINALES			
Descripción	Uds.	Valor unitario	Total
Costos Compras	1	840,00 €	840,00 €
Costos material	1	71.749,49 €	71.749,49 €
Costos mano obra mecánica	1	27.140,70 €	27.140,70 €
Costos mano obra eléctrica	1	9.072,00 €	9.072,00 €
Costos Ingeniería (Diseño planos eléctricos, mecánicos y automatización)	1	22.390,50 €	€26.845,50€
TOTAL			135.647,69 €
BENEFICIO 30%			40.694,31 €
COSTO TOTAL INSTALACION			176.342,00 €

Tabla 27 Costes finales

8. Conclusiones

Podemos decir que el resultado final del proyecto ha sido satisfactorio. Se han alcanzado los objetivos marcados de la manera acordada y en el tiempo planificado. Se han conseguido los hitos de funcionamiento en las dos líneas. Hemos logrado una máquina funcional y que mejora la capacidad de paletizado de la instalación.

En este tipo de instalaciones es muy importante la puesta en marcha del sistema. Requiere poner especial atención la instalación del robot, pues deben medirse bien la situación de los elementos y del vallado antes de anclar la peana sobre la que apoyará el robot.

El robot KR 180 3200 PA resulta ser un robot idóneo para paletización. Se han obtenido unas velocidades y aceleraciones óptimas en cada uno de los movimientos. Ha habido que limitar en cierta medida la velocidad de funcionamiento del robot porque en ocasiones generaba unas inercias que, al transportar un palet, podían ocasionar daños a la instalación.

El software de programación del robot KUKA es muy potente y le diferencia claramente de otras marcas de robots. Esto se debe a que utiliza un PC como unidad de control, otorgándole así una mayor potencia de cálculo y procesamiento. Por lo tanto la sintaxis se aleja de la tradicional de robot y se asemeja más a un lenguaje de alto nivel.

Personalmente durante la realización del proyecto aprendí muchísimo sobre la gestión de equipos y la programación de la robótica industrial.

9. Bibliografía y referencias

- KUKA KRC4 Instrucciones de servicio KRC4. Edición: 22.05.2012. Versión: BA KR C4 GI V5
- KUKA INFORMACIÓN BÁSICA EXTENDIDA instrucciones de servicio y programación para los integradores de sistemas. Edición: 23.01.2014. Versión: KSS 8.3 SI V2
- KUKA KR QUANTEC PA Instrucciones de servicio. Edición: 08.11.2012. Versión BA KR QUANTEC PA V3
- El libro blanco de la robótica en España: Investigación, tecnologías y formación Editorial CEA
- PFC ROBOT PALETIZADOR DE PRODUCTO HETEROGÉNEO MARCA MOTOMAN. DOS LÍNEAS DE COGIDA Y DOS LÍNEAS DE DEJADA. Autor: Antonio Martin Blanco

Recursos web:

- http://www.tecnowey.com/media/catalogos/por%20productos/transporte_pallets_y_cabeceras_almacen.pdf Ultimo acceso: 18.02.2016
- <http://www.tecnowey.com/media/catalogos/generales/tecnowey%20PALETIZADO%20PDF%20digital.pdf> Ultimo acceso: 18.02.2016
- https://alojamientos.uva.es/guia_docente/uploads/2012/372/50004/1/Documento1.pdf Ultimo acceso: 18.02.2016
- <http://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/log%C3%ADstica/paletizaci%C3%B3n/> Ultimo acceso: 18.02.2016

10. ANEXOS

PLANOS DE LA INSTALACIÓN

Alimentacion: **III+N+T.T. 400V**

Potencia Instalada: **20KW (100A)**

Tension Mando 1: **230VAC**

Tension Mando 2: **24VDC**

Tension Mando 3:

Tipo PLC: **CJ2**

Circuito de potencia: **NEGRO**

Conductor de neutro: **AZUL**

Conductor de toma tierra: **AMARILLO/VERDE**

Circuito de mando 220Vac: **ROJO**

Circuito de mando 24Vac:

Circuito de mando 24Vdc: **AZUL**

Hilo tension exterior: **NARANJA**

INDICE	MODIFICACION	FECHA	DIBUJADO	VERIFICADO	APROBADO

**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

Av. de la Estacion, 12 - Pol. Ind. "LAS CANTERAS"
45520 VILLALUENGA DE LA SAGRA (TOLEDO)

TF +34 925537970

info@asasl.es

FAX: +34 925537954

DIBUJADO:

VERIFICADO:

FECHA DE CREACION: Octubre 2015

PROYECTO N°: 15111211

PALETIZADO ROLLOS DE PAPEL

CON ROBOT

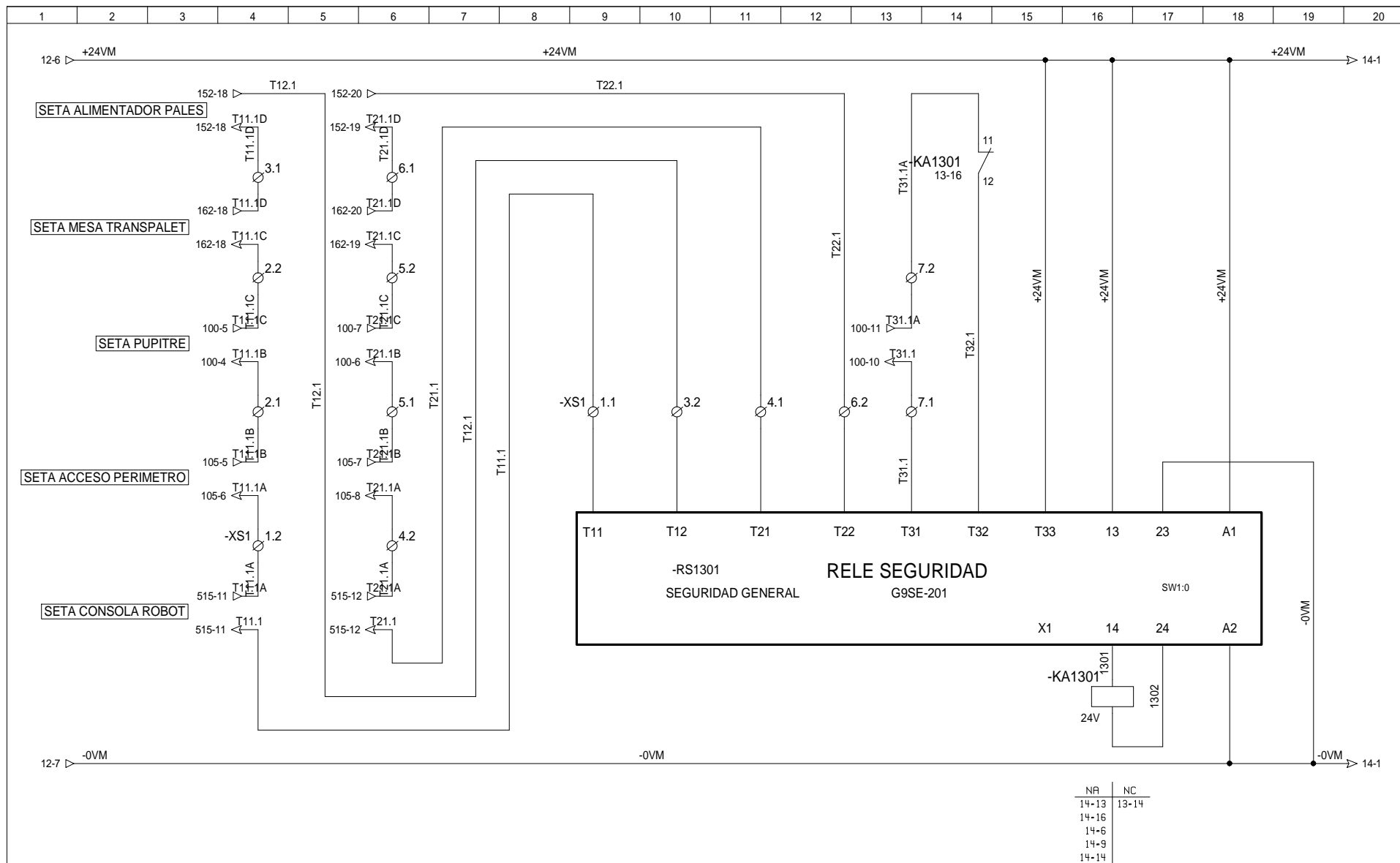
ESQUEMAS ELECTRICOS

PORTADA

DOCUMENTO N° : **15111211**

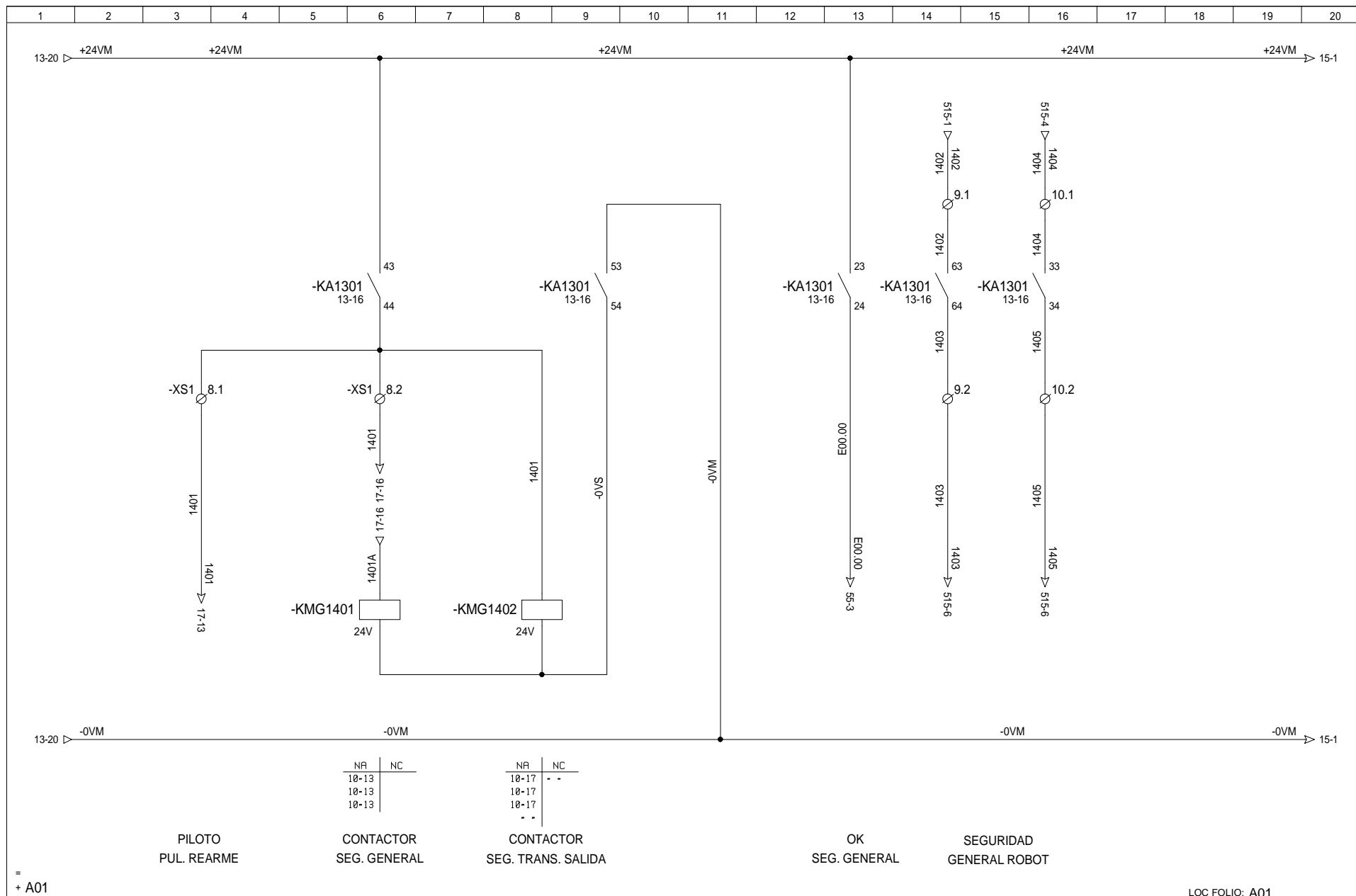
1 / 59

ESTE DOCUMENTO ES PROPIEDAD DE AUTOMATIZACION Y SOLUCIONES AVANZADAS.
ESTA PROHIBIDA TODA REPRODUCCION TOTAL O PARCIAL SIN AUTORIZACION.



=
+ A01

LOC FOLIO: A01



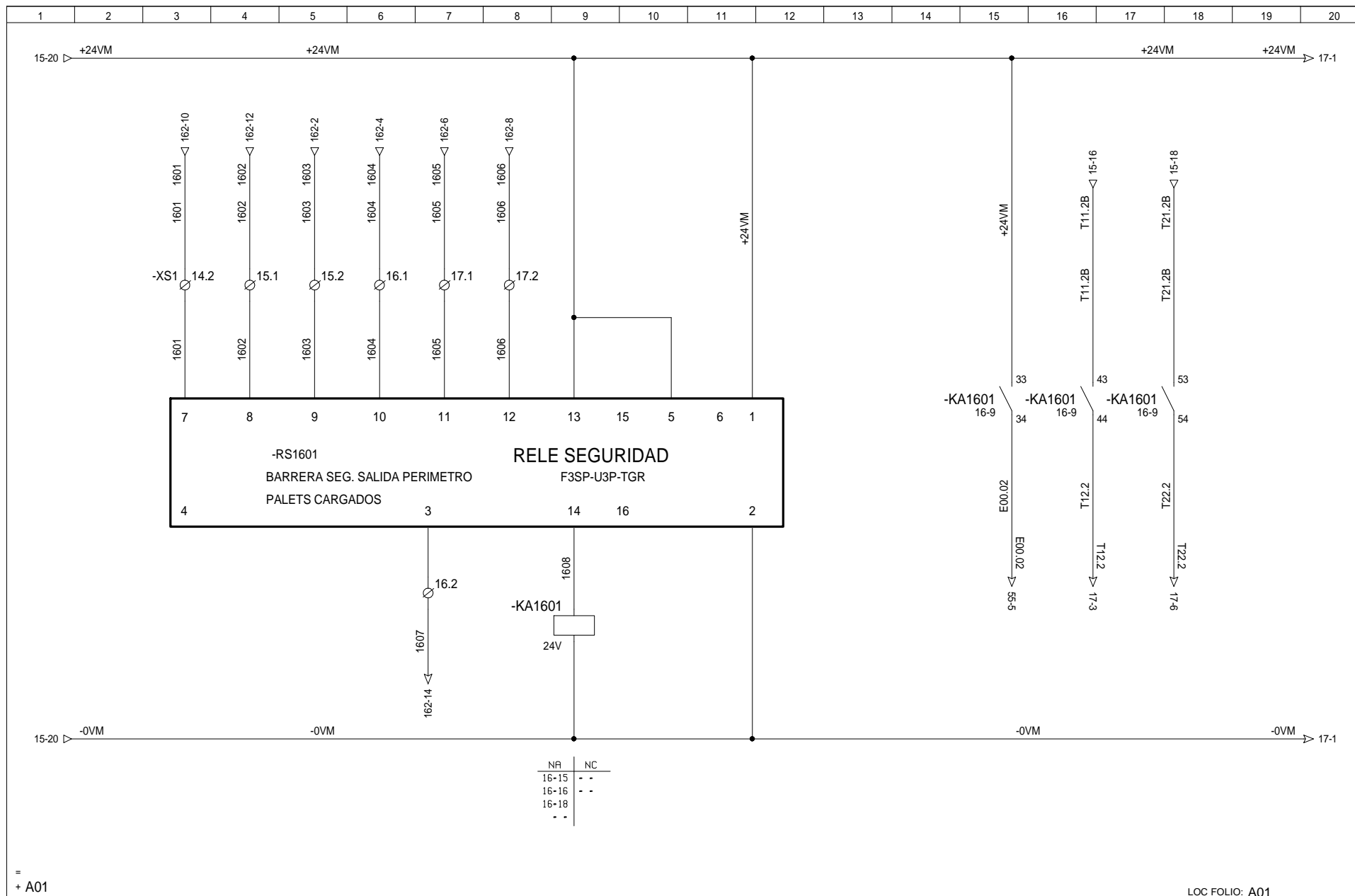
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

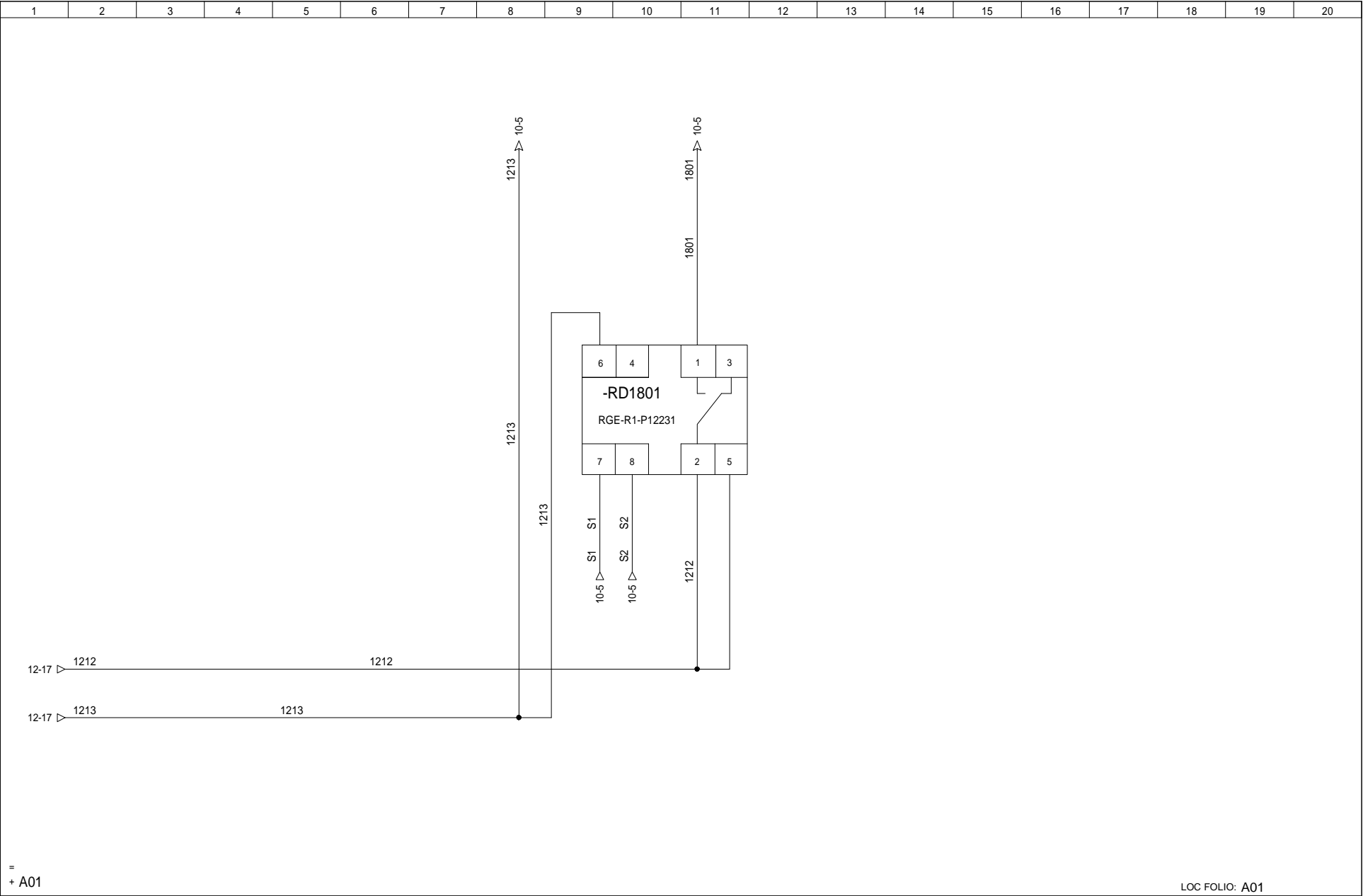
DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
SEGURIDAD GENERAL
LINEA DE PALETIZADO

FOLIO
14
◀ 13 15 ▶





=
+ A01

LOC FOLIO: A01



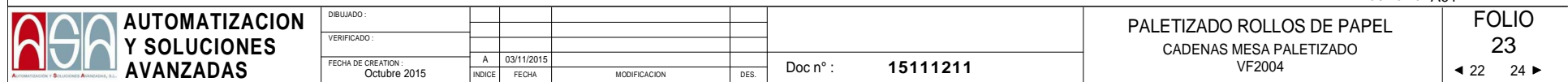
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

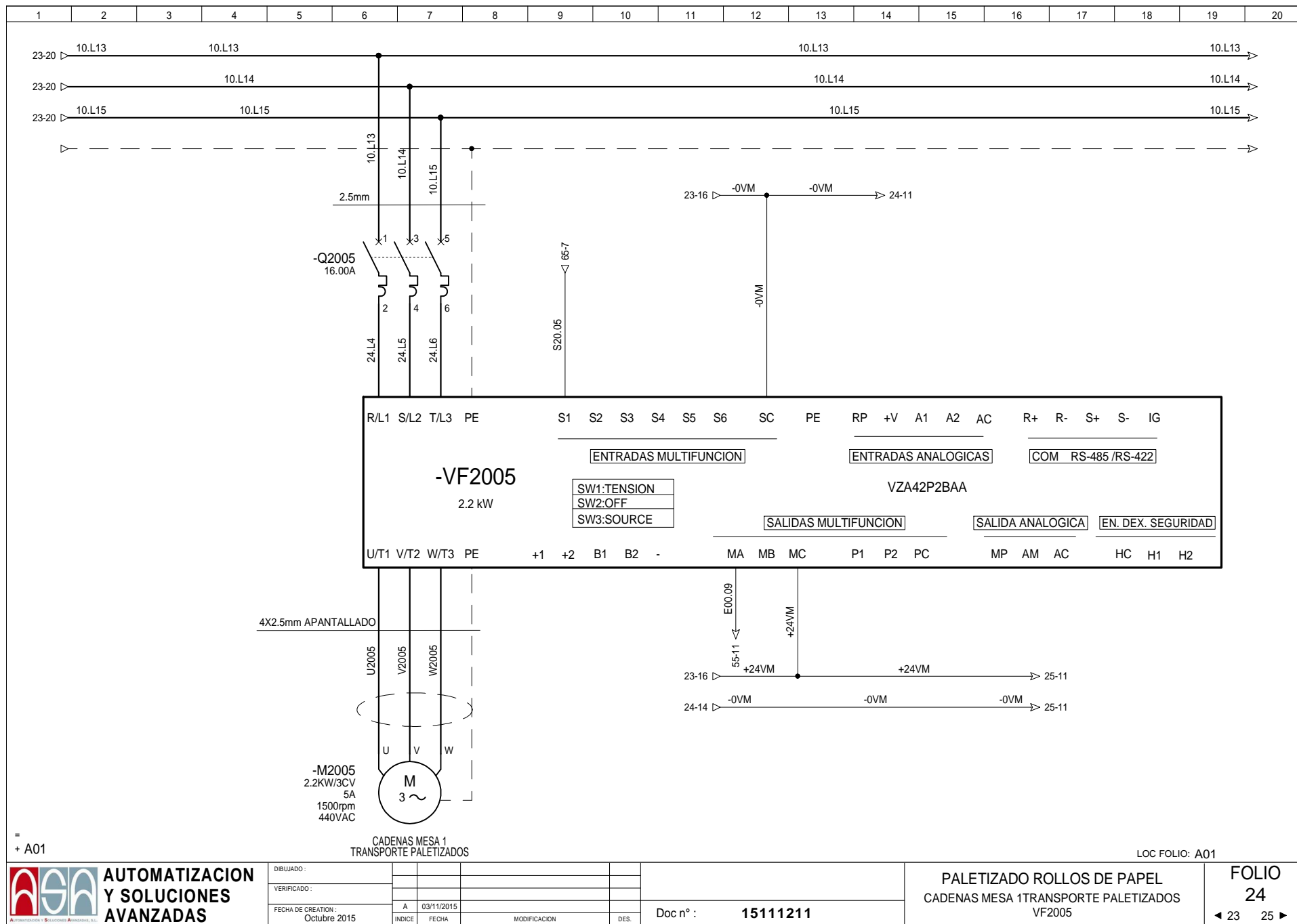
DIBUJADO :					
VERIFICADO :					
FECHA DE CREATION :	A	03/11/2015			
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.	

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
PROTECCION DIFERENCIAL
CONEXIONES

FOLIO
18
◀ 17 20 ▶





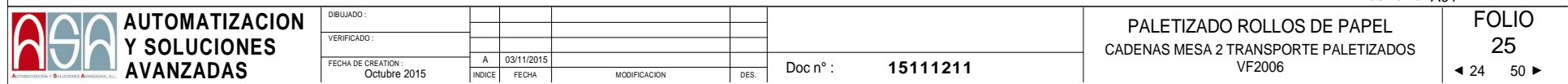
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

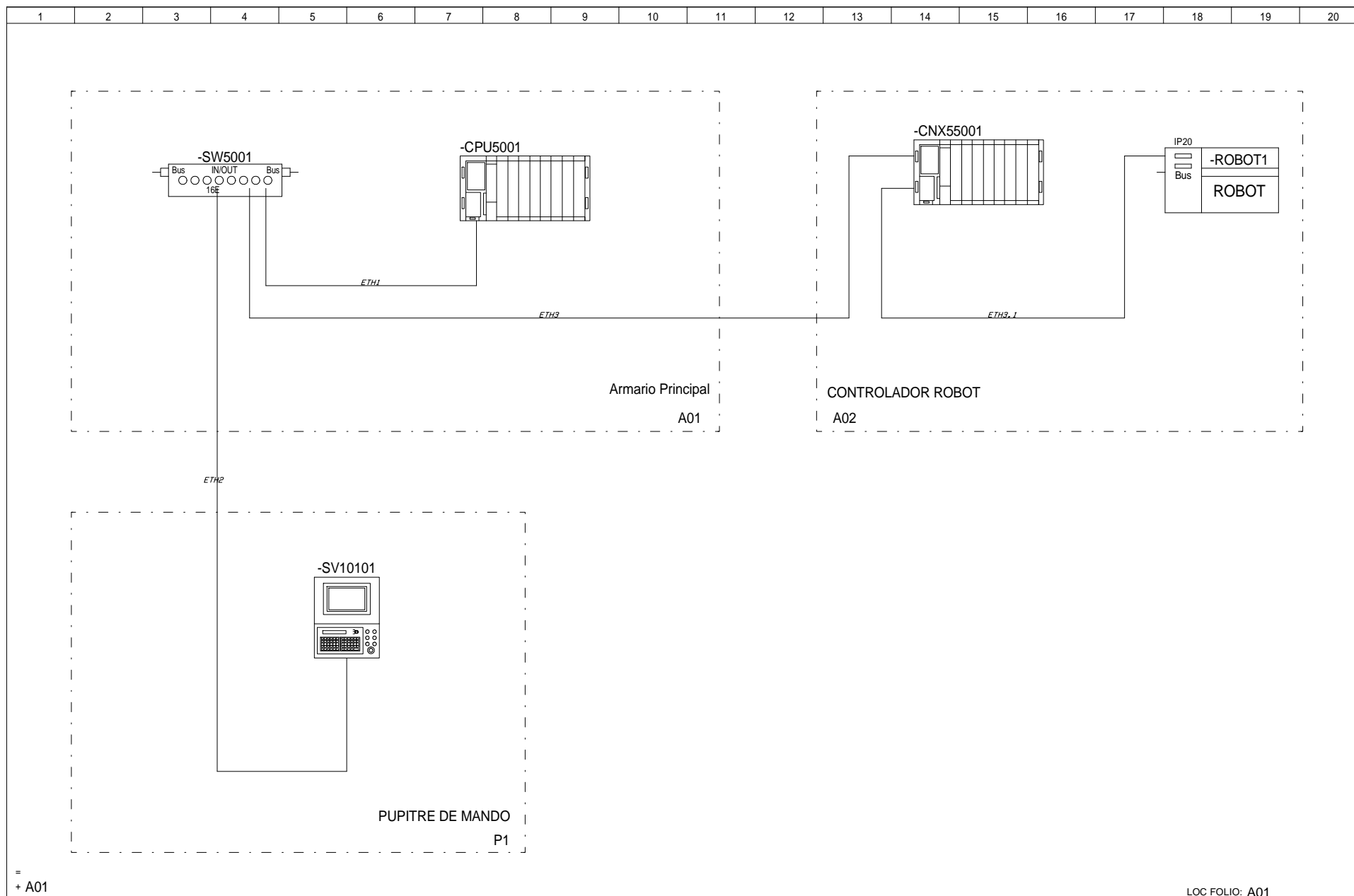
DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

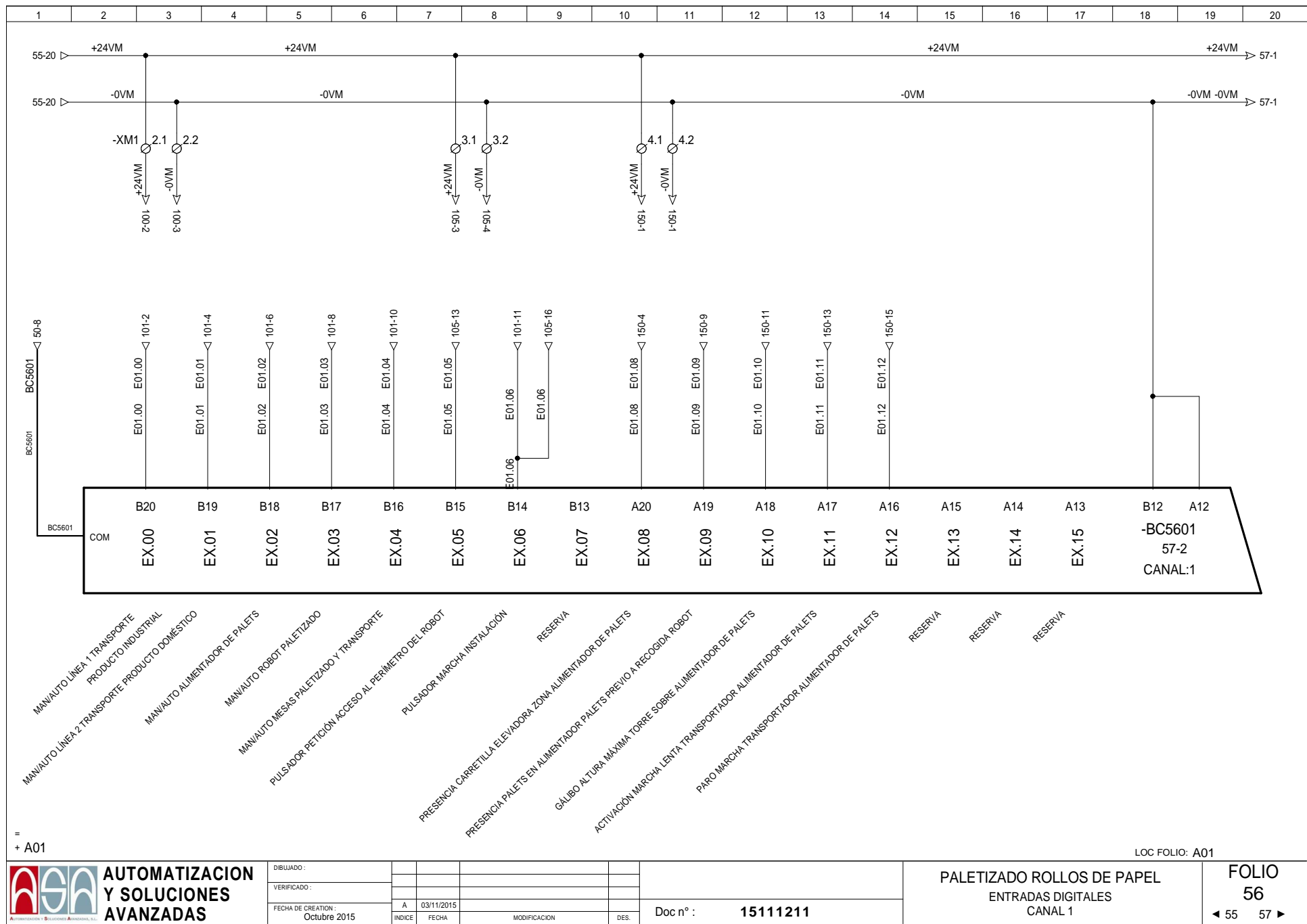
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CADENAS MESA 1TRANSPORTE PALETIZADOS
VF2005

FOLIO
24
◀ 23 25 ▶







**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

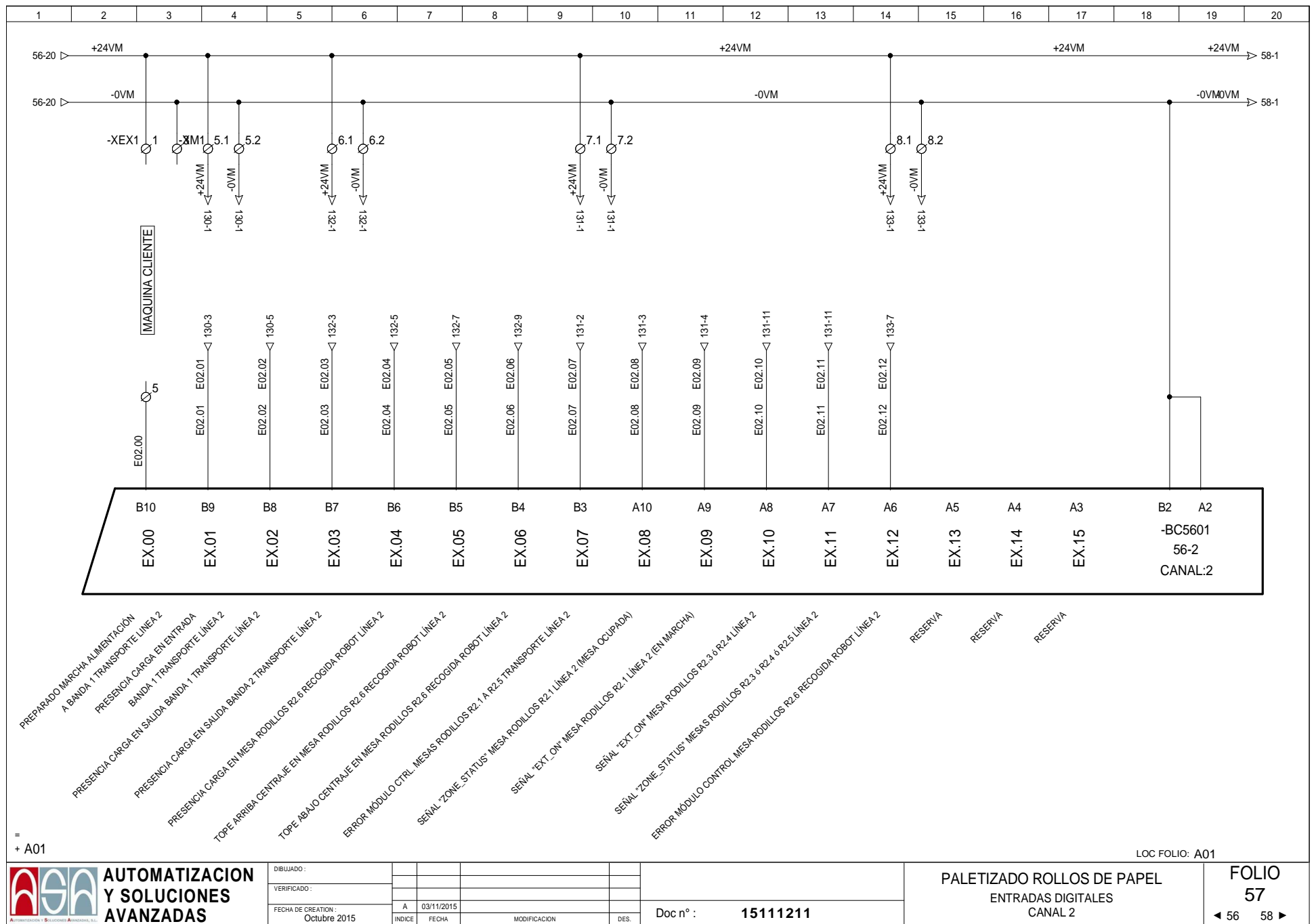
DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
ENTRADAS DIGITALES
CANAL 1

LOC FOLIO: A01

FOLIO
56
◀ 55 57 ▶



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
INDICE	FECHA	MODIFICACION	DES.	

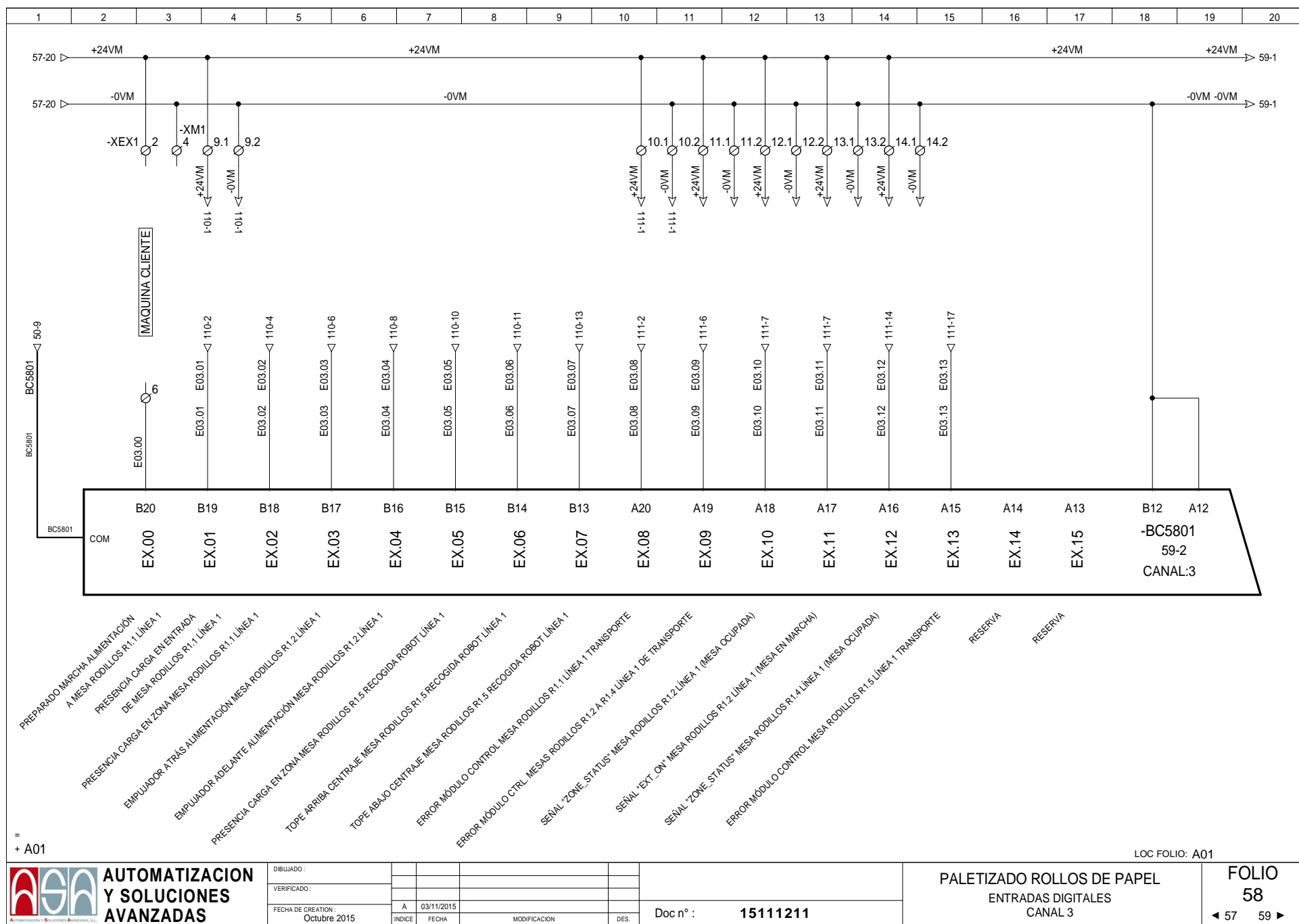
Doc n° : **15111211**

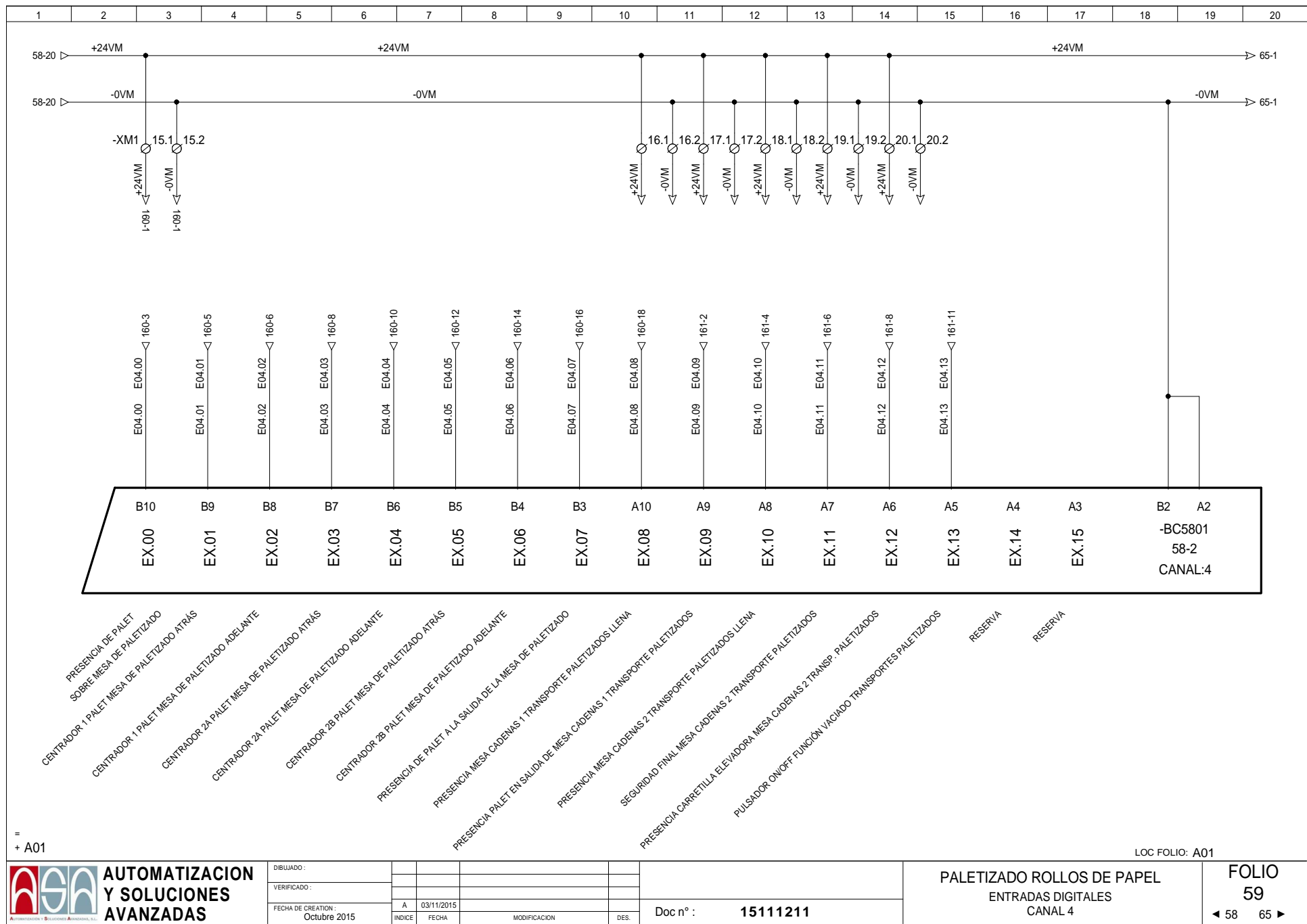
PALETIZADO ROLLOS DE PAPEL
ENTRADAS DIGITALES
CANAL 2

LOC FOLIO: A01

FOLIO
57

◀ 56 58 ▶





**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

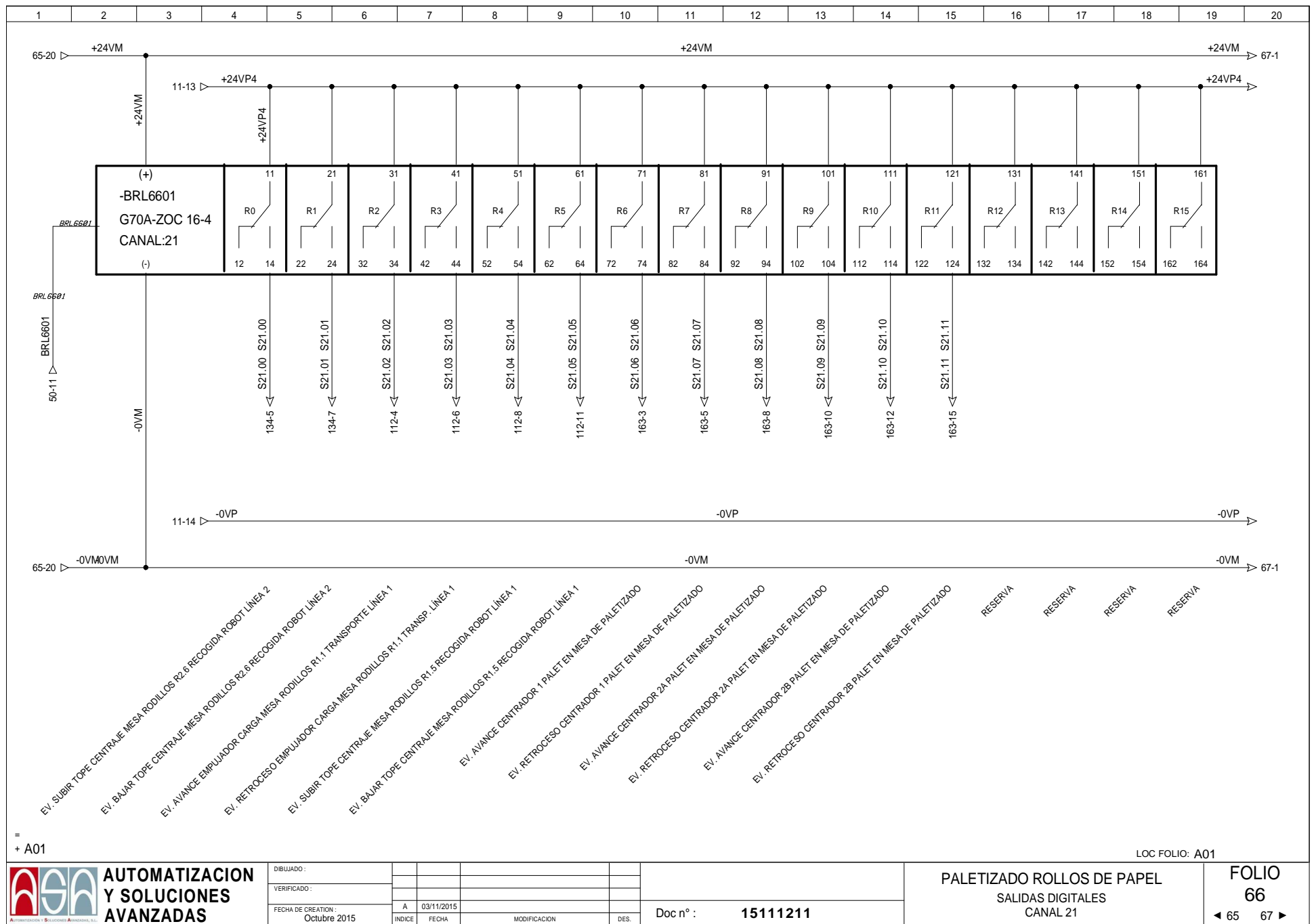
Doc n° : **15111211**

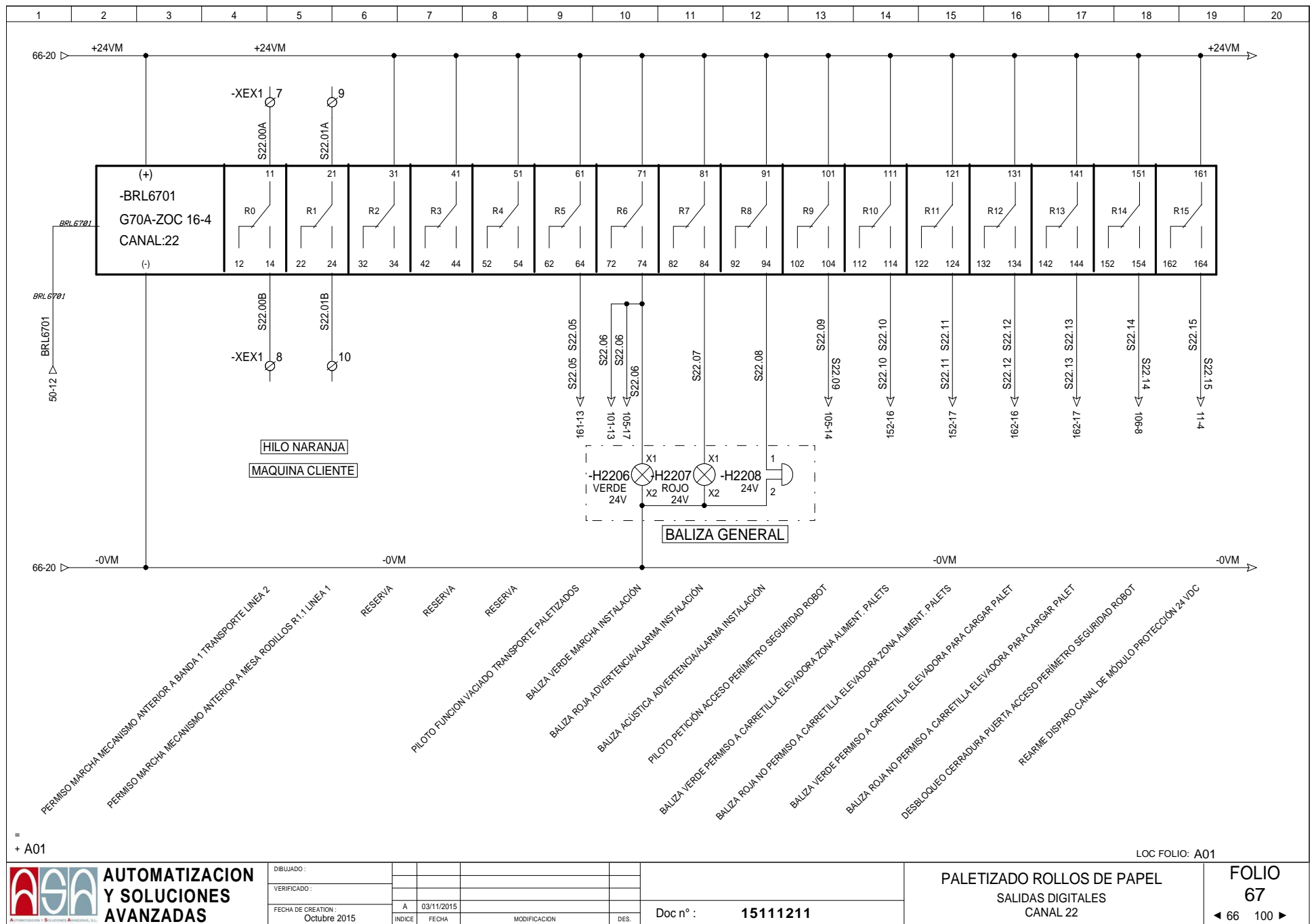
PALETIZADO ROLLOS DE PAPEL
ENTRADAS DIGITALES
CANAL 4

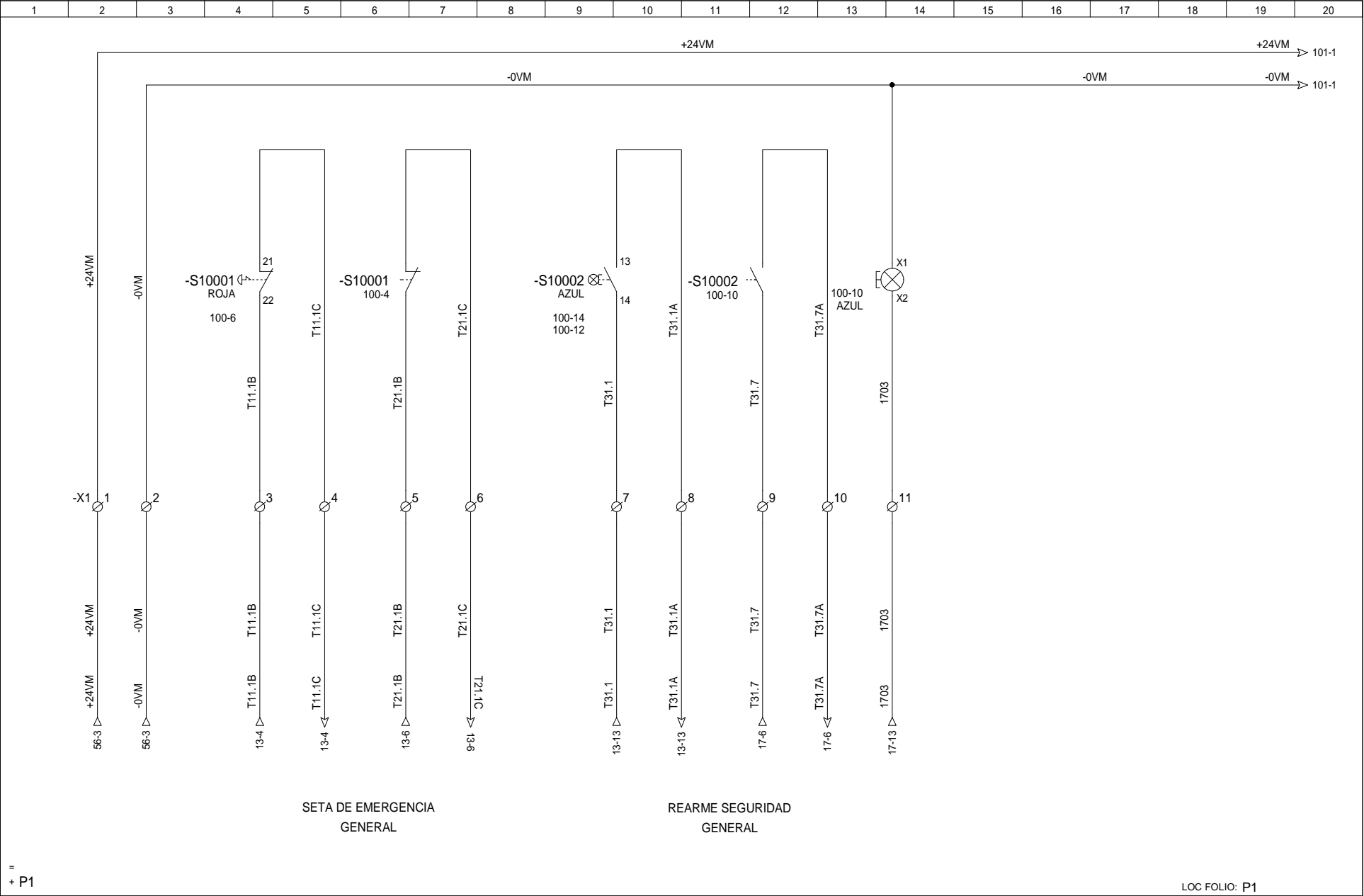
LOC FOLIO: A01

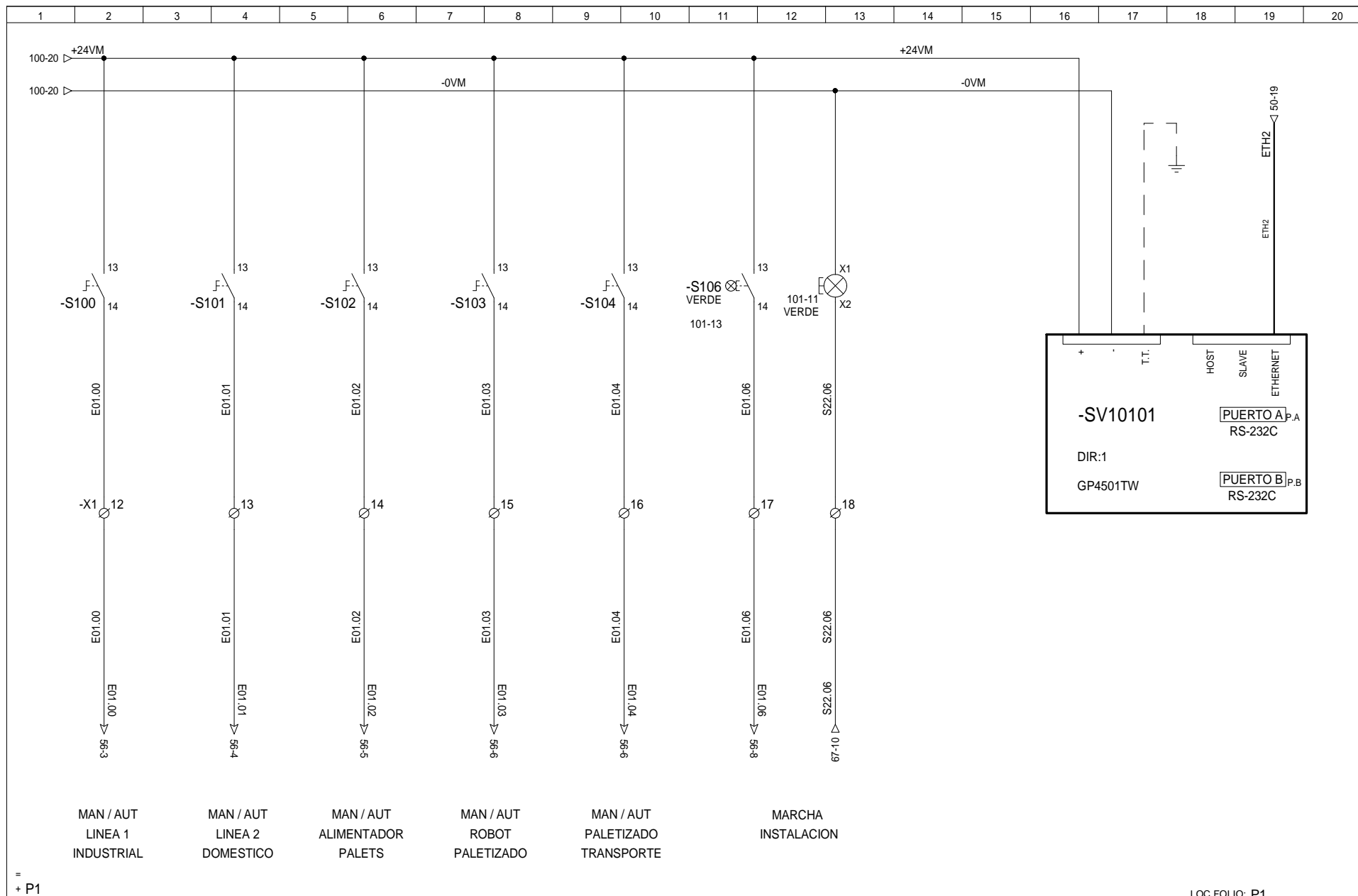
FOLIO
59

◀ 58 65 ▶

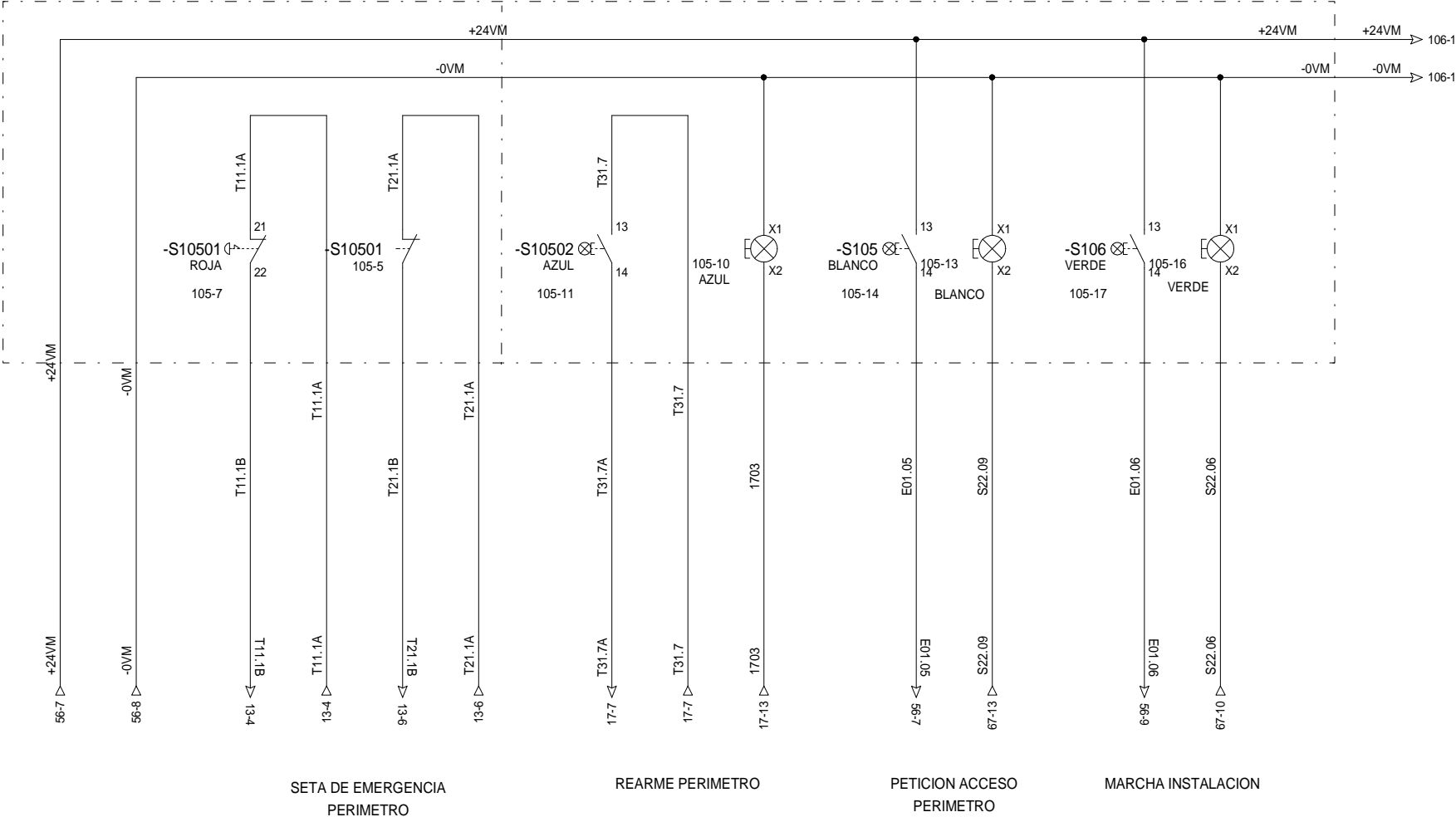








B1 BOTONERA ACCESO PERIMETRO



=
+ B1

LOC FOLIO: B1



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

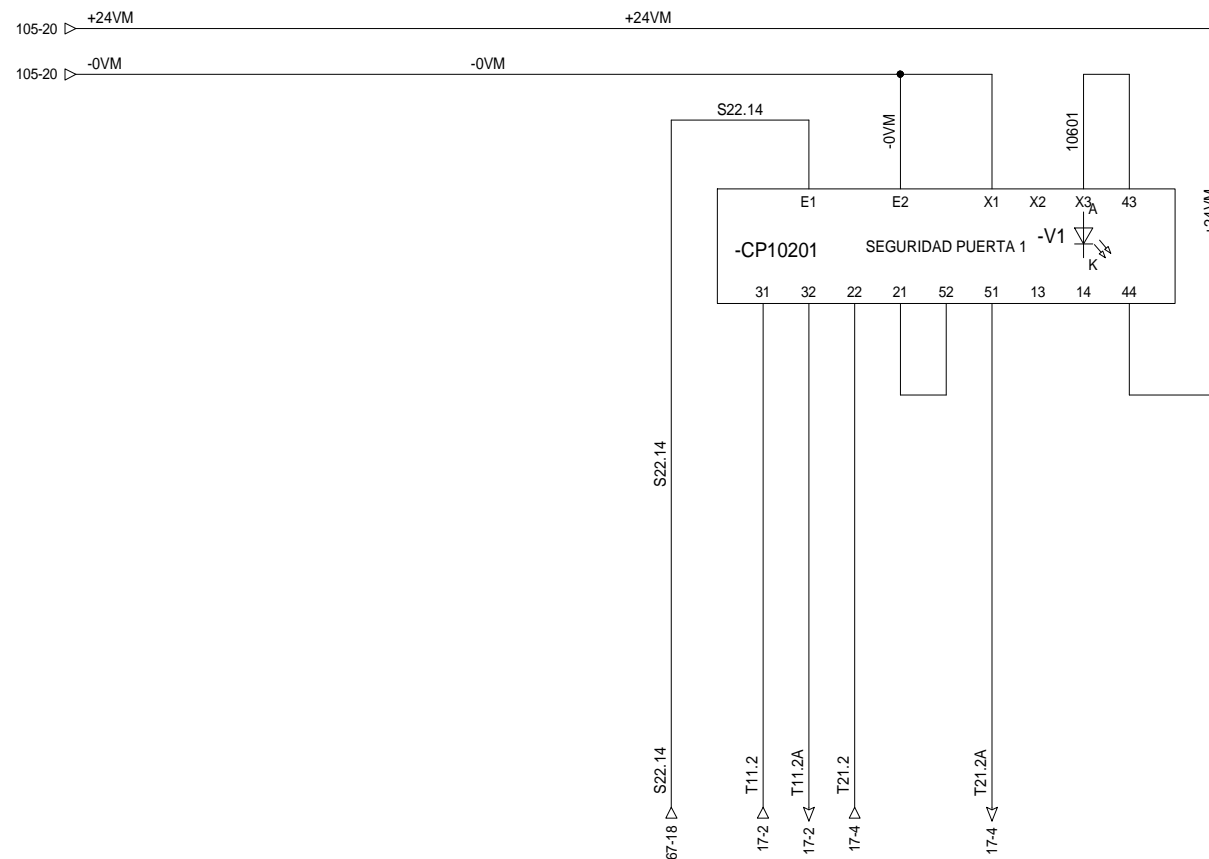
DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

**PALETIZADO ROLLOS DE PAPEL
BOTONERA ACCESO PERIMETRO (B1)
CONEXIONES**

**FOLIO
105**
◀ 101 106 ▶

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



=
+ B1

LOC FOLIO: B1



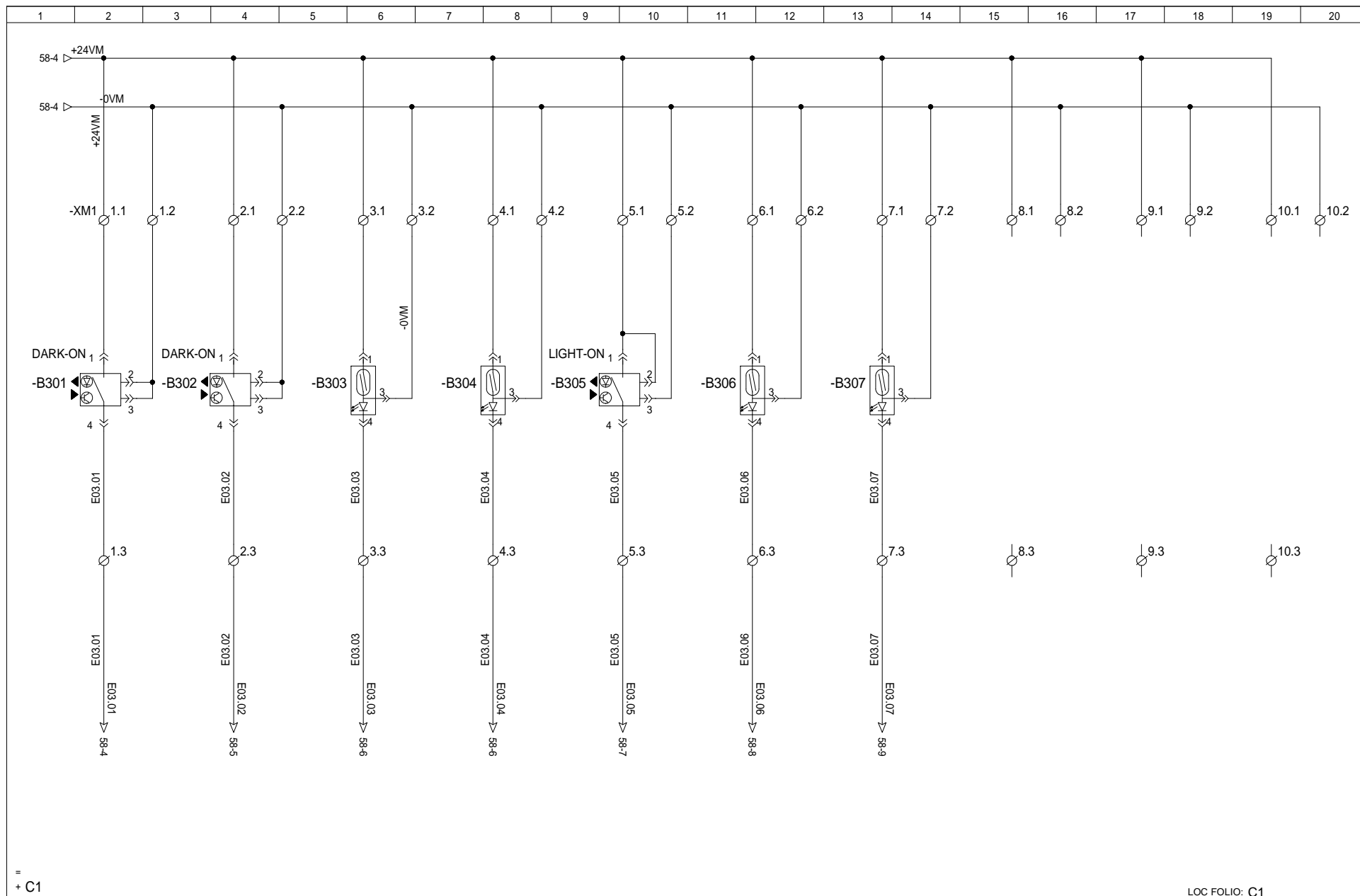
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO :					
VERIFICADO :					
FECHA DE CREATION :	A	03/11/2015			
INDICE	FECHA	MODIFICACION	DES.		

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
INTERRUPTOR DE SEGURIDAD
PUERTA DE ACCESO

FOLIO
106
◀ 105 110 ▶



LOC FOLIO: C1



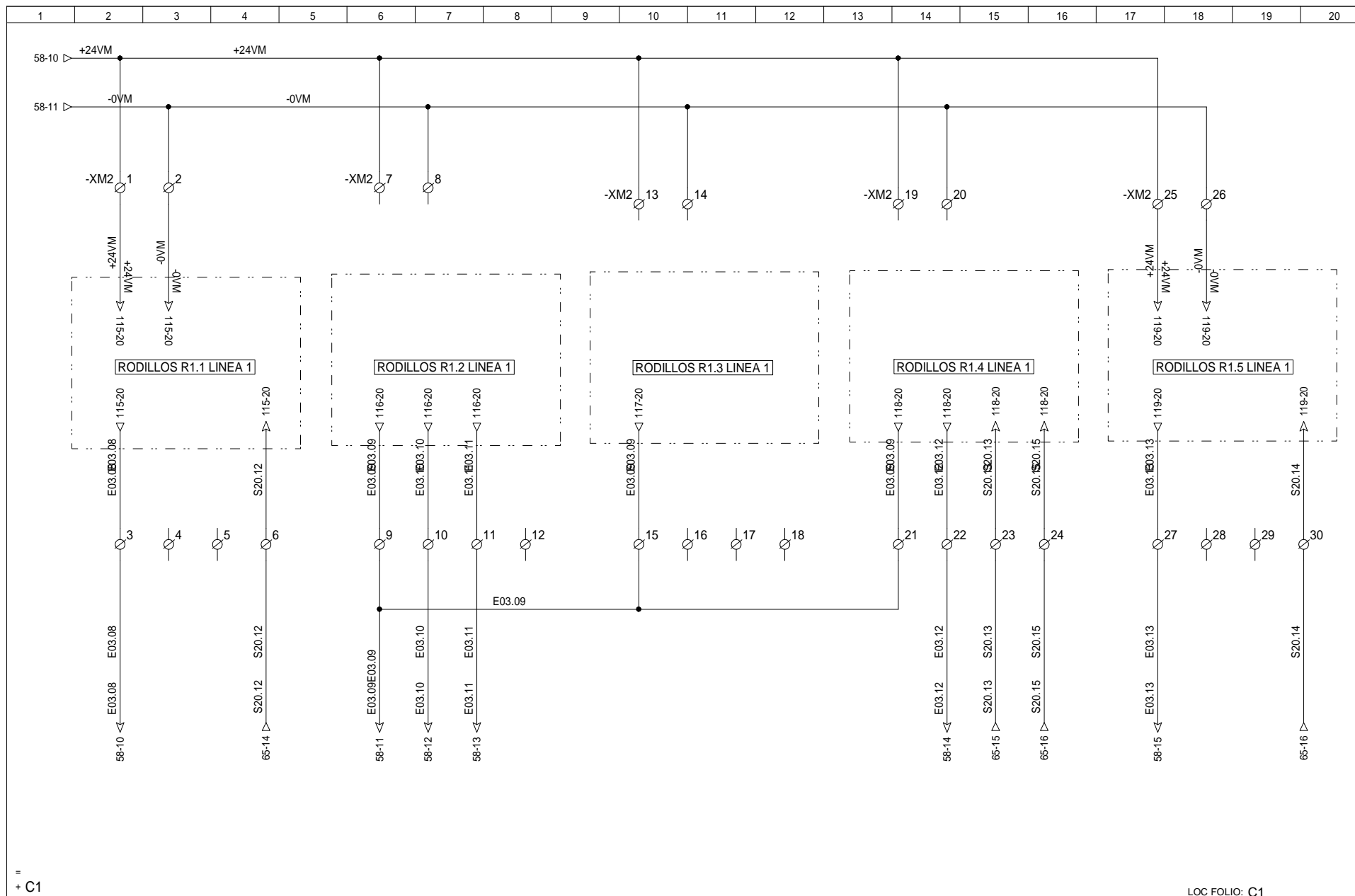
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREACION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 1
LINEA 1 TRANSPORTE

FOLIO
110
◀ 106 111 ▶



=
+ C1

LOC FOLIO: C1



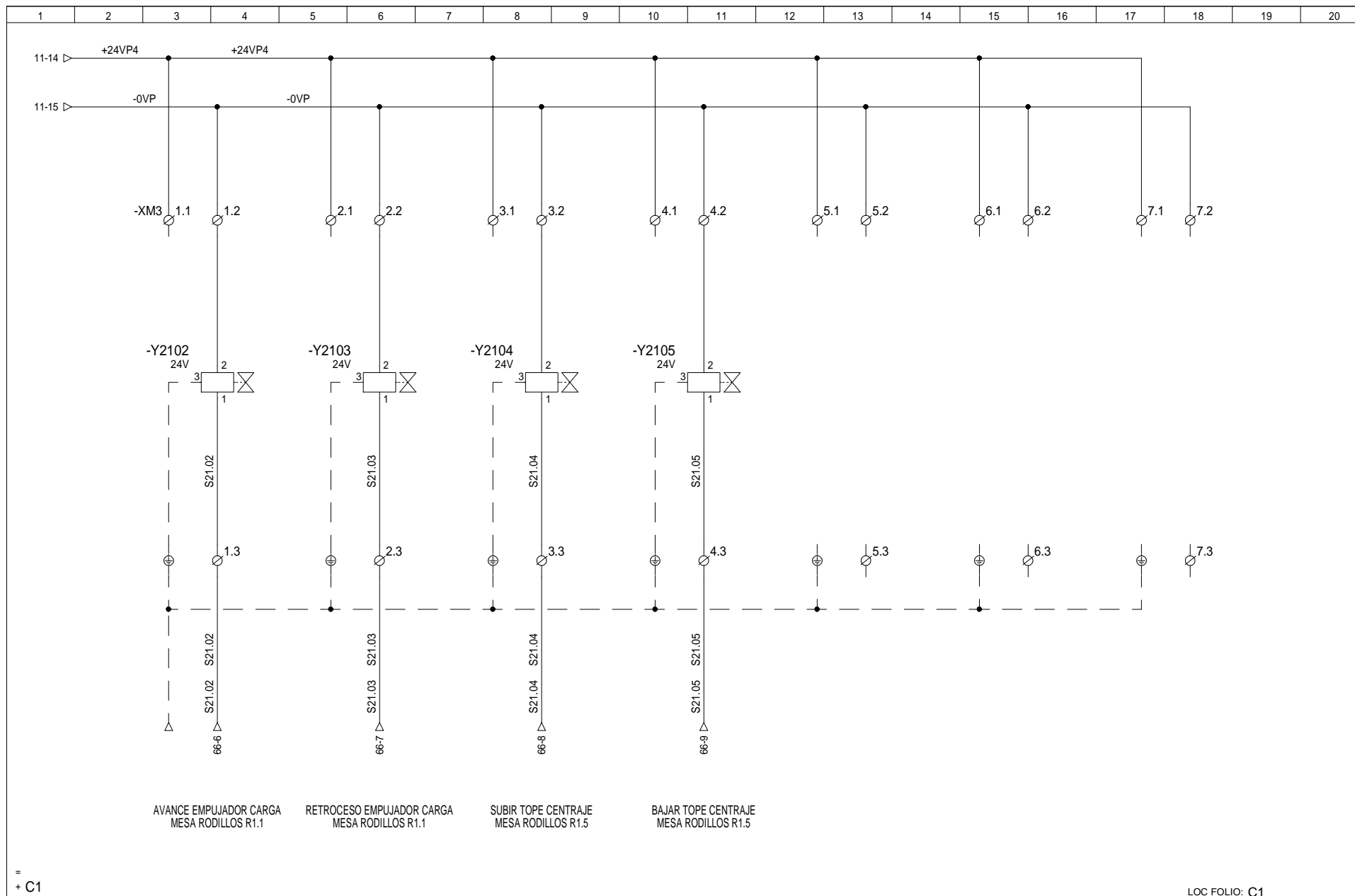
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREACION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 1
LINEA 1 TRANSPORTE

FOLIO
111
◀ 110 112 ▶



LOC FOLIO: C1



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

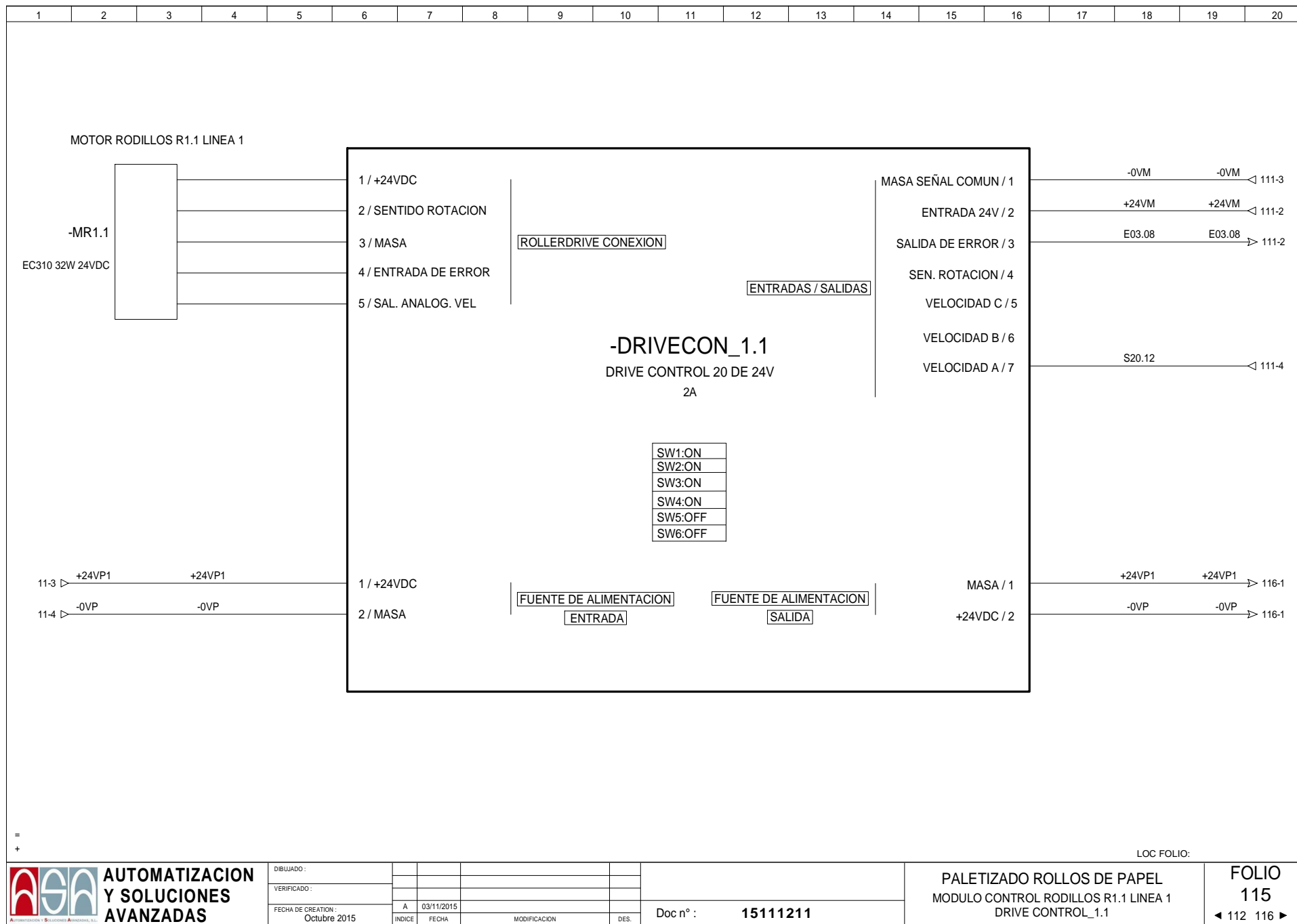
DIBUJADO:				
VERIFICADO:				
FECHA DE CREACION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

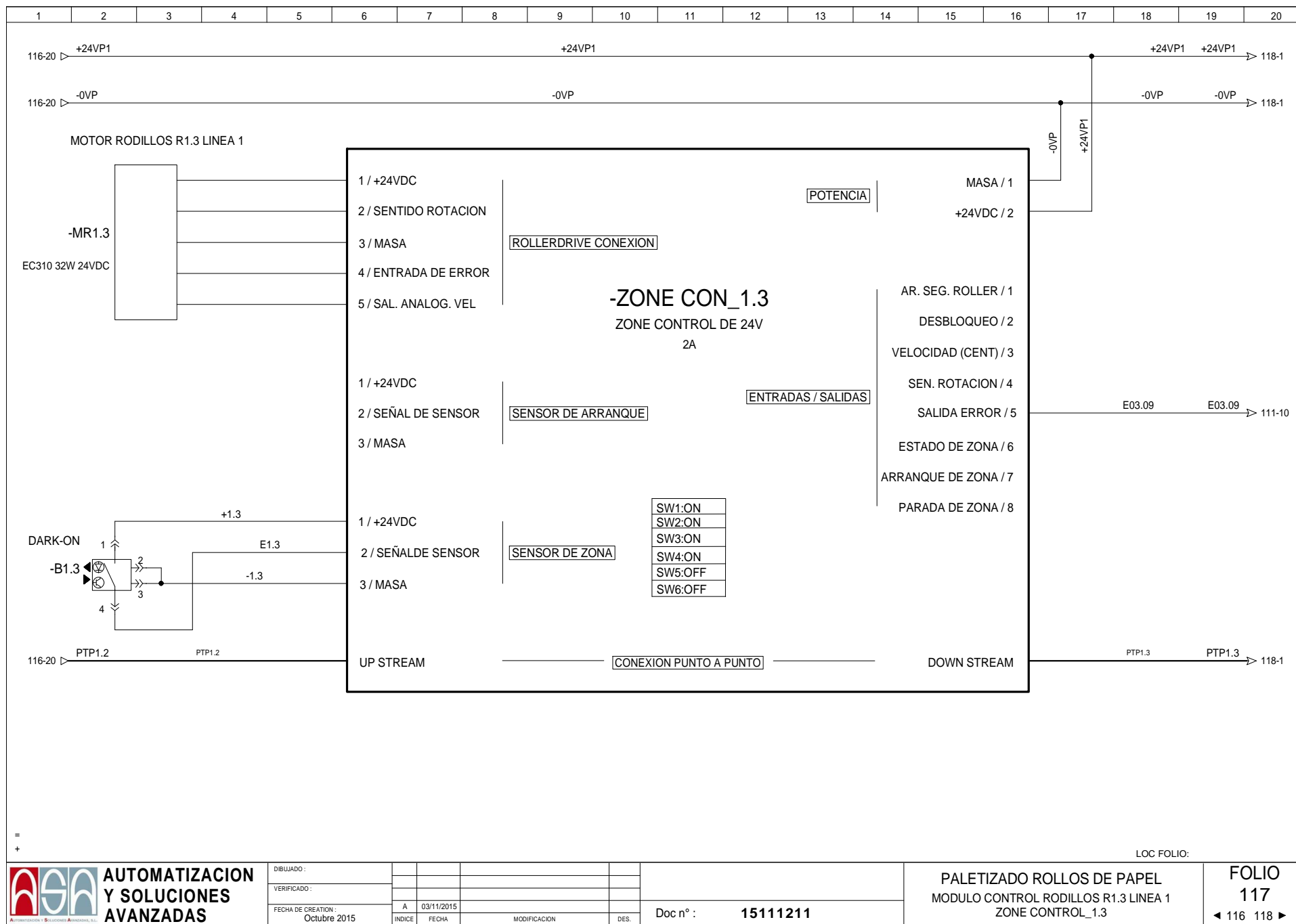
PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 1
LINEA DE TRANSPORTE 1

**FOLIO
112**

◀ 111 115 ▶







**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

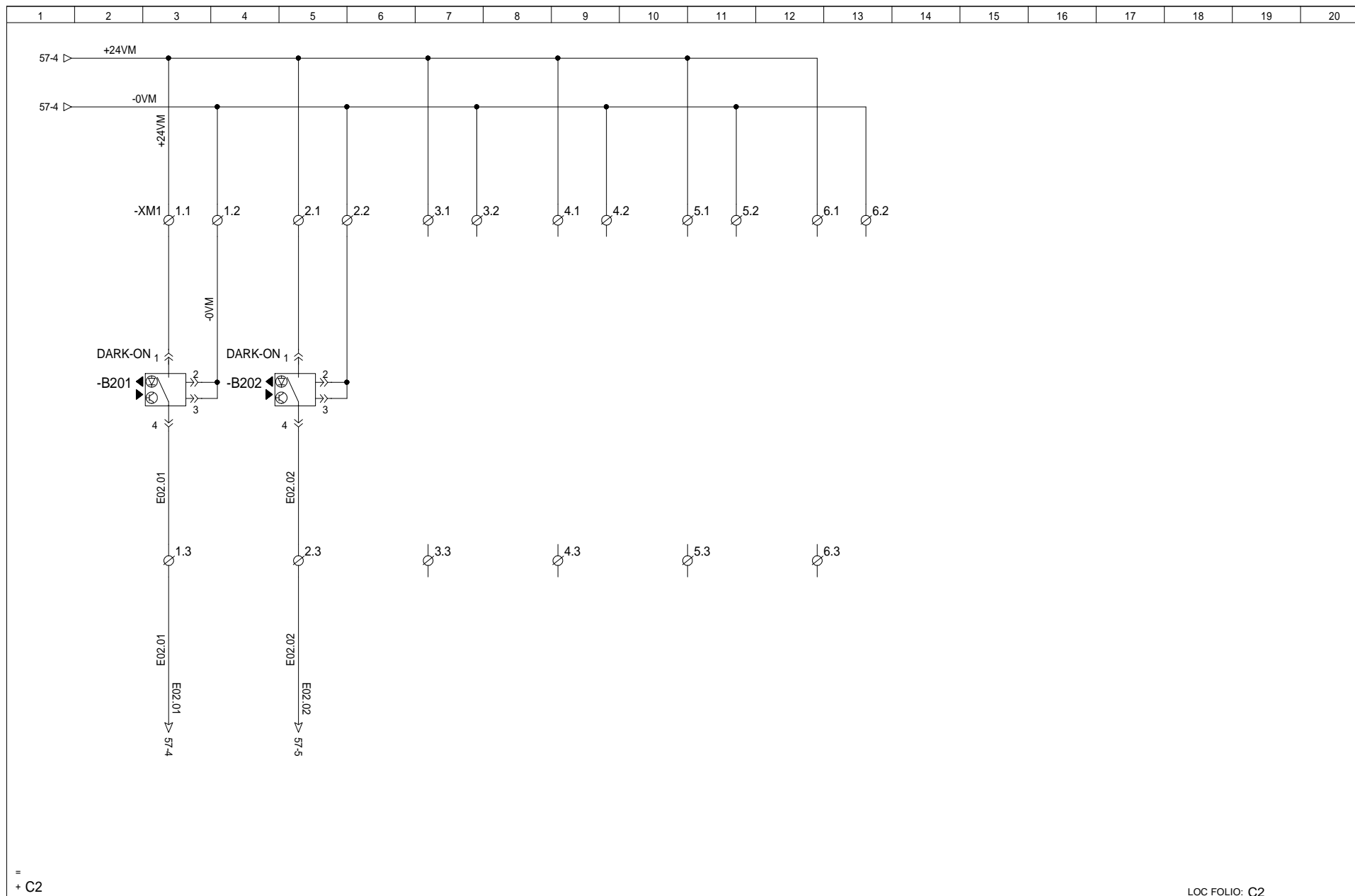
Doc n° : **15111211**

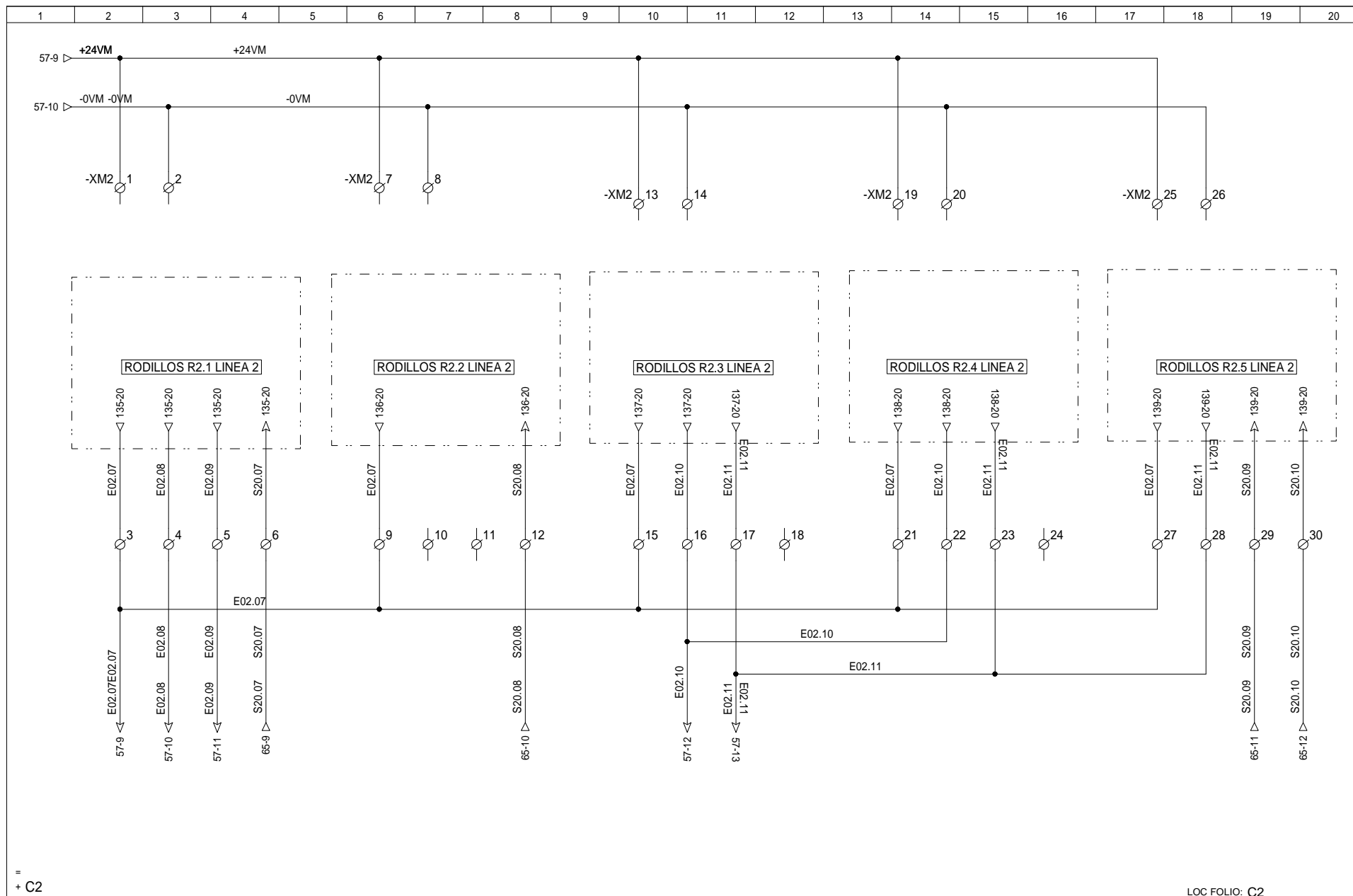
PALETIZADO ROLLOS DE PAPEL
MODULO CONTROL RODILLOS R1.3 LINEA 1
ZONE CONTROL_1.3

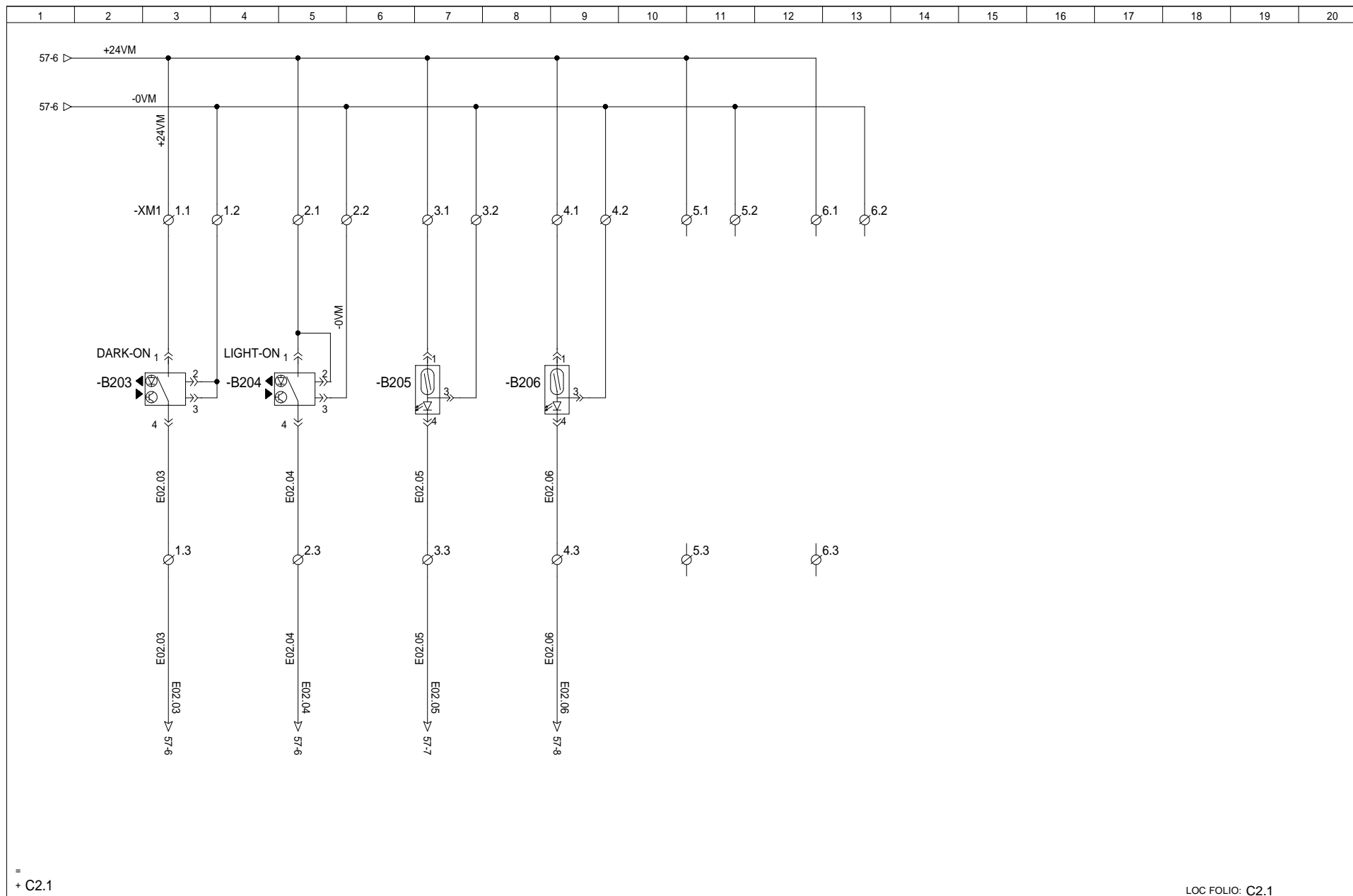
LOC FOLIO:

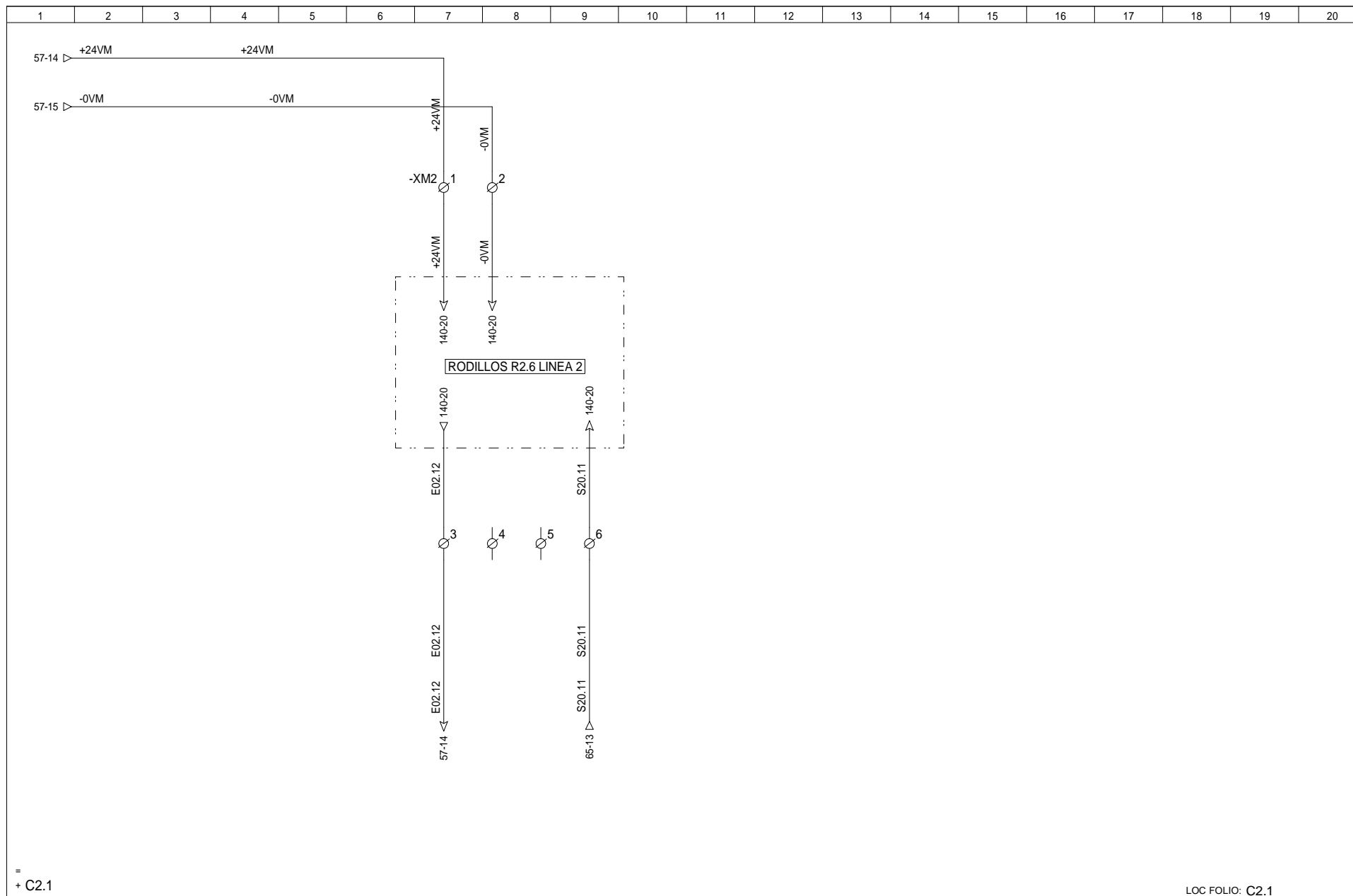
**FOLIO
117**

◀ 116 118 ▶









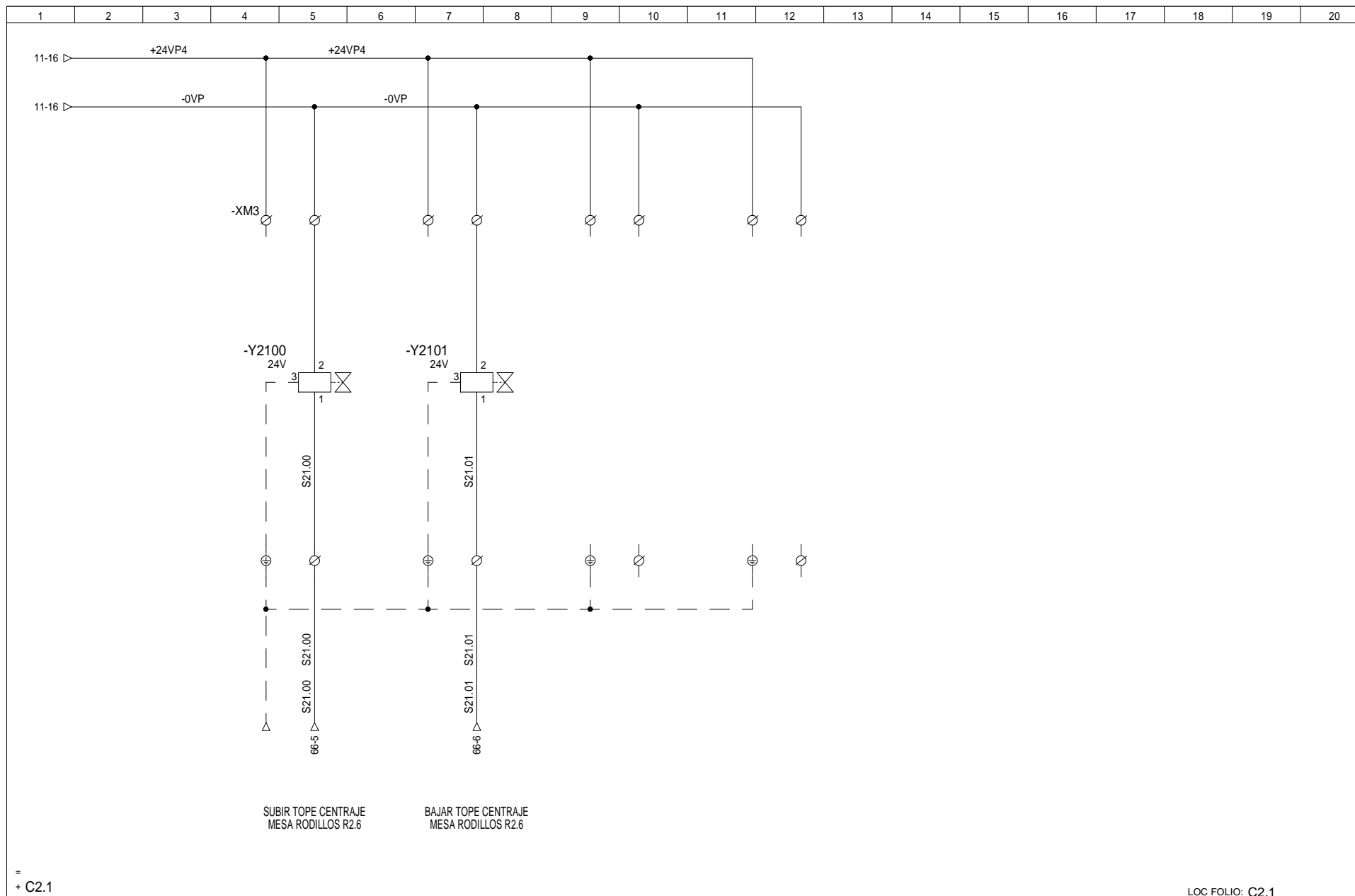
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

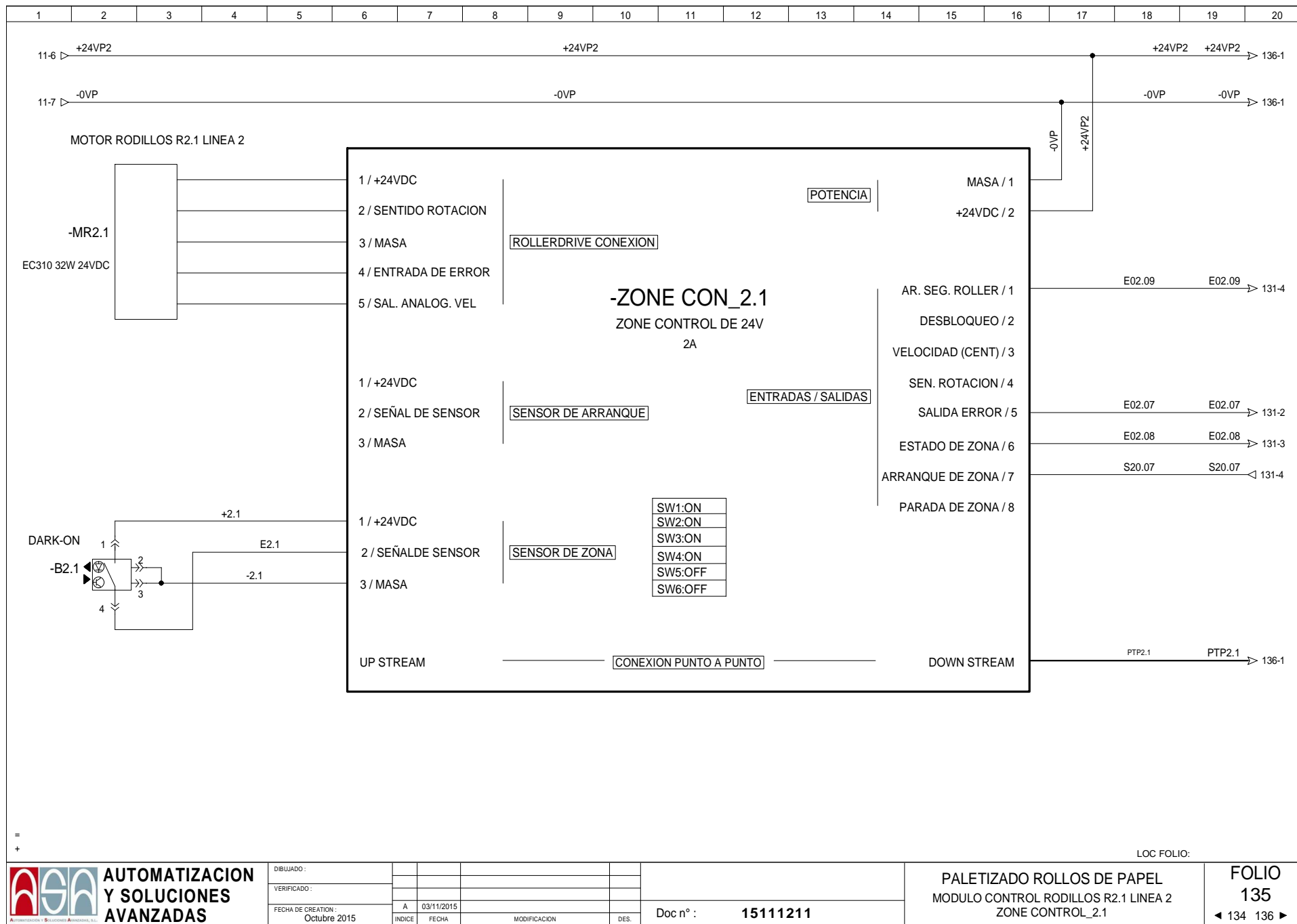
DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 2.1
LINEA DE TRANSPORTE 2 R2.6

FOLIO
133
◀ 132 134 ▶





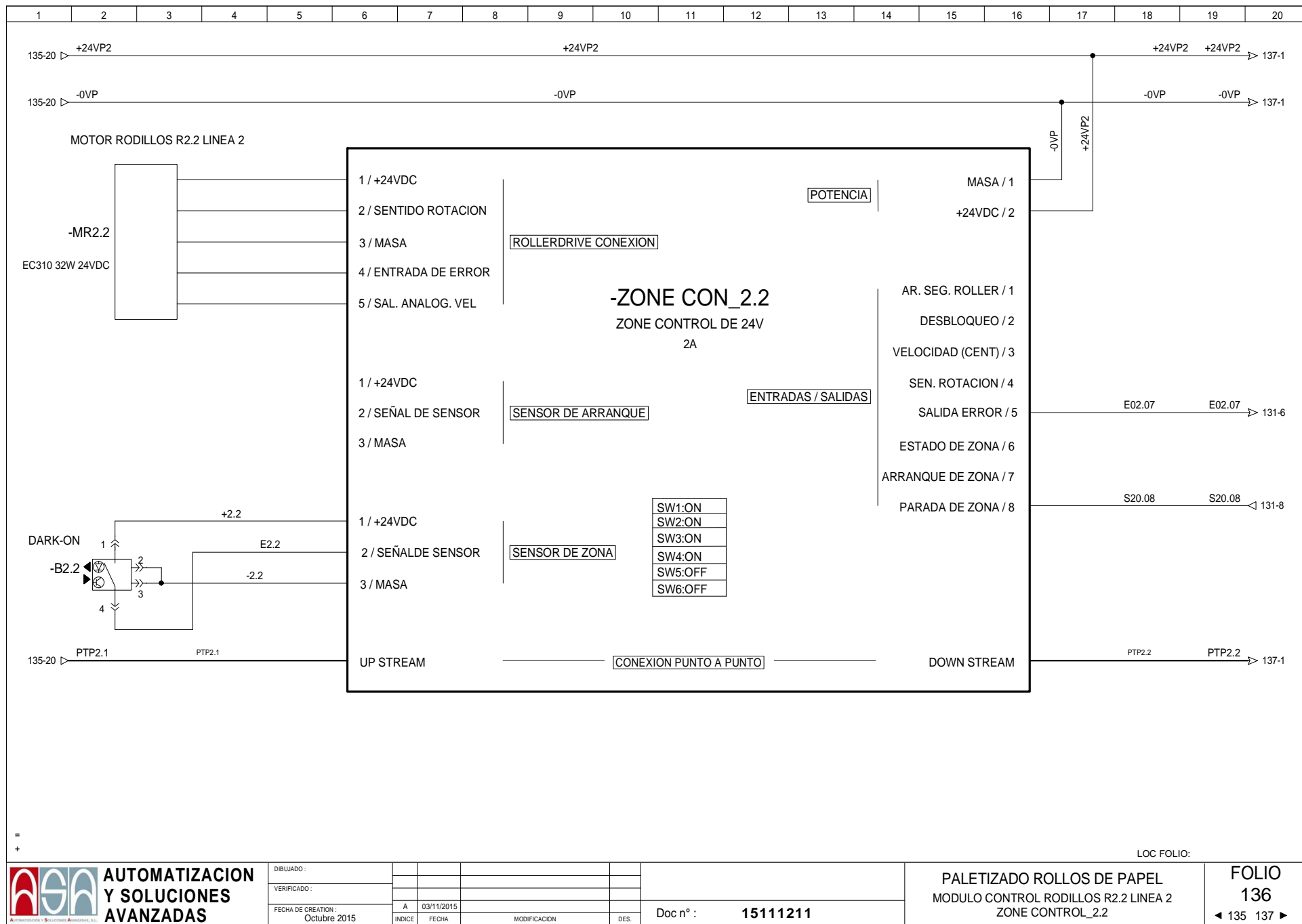
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
MODULO CONTROL RODILLOS R2.1 LINEA 2
ZONE CONTROL_2.1

LOC FOLIO:
**FOLIO
135**
◀ 134 136 ▶



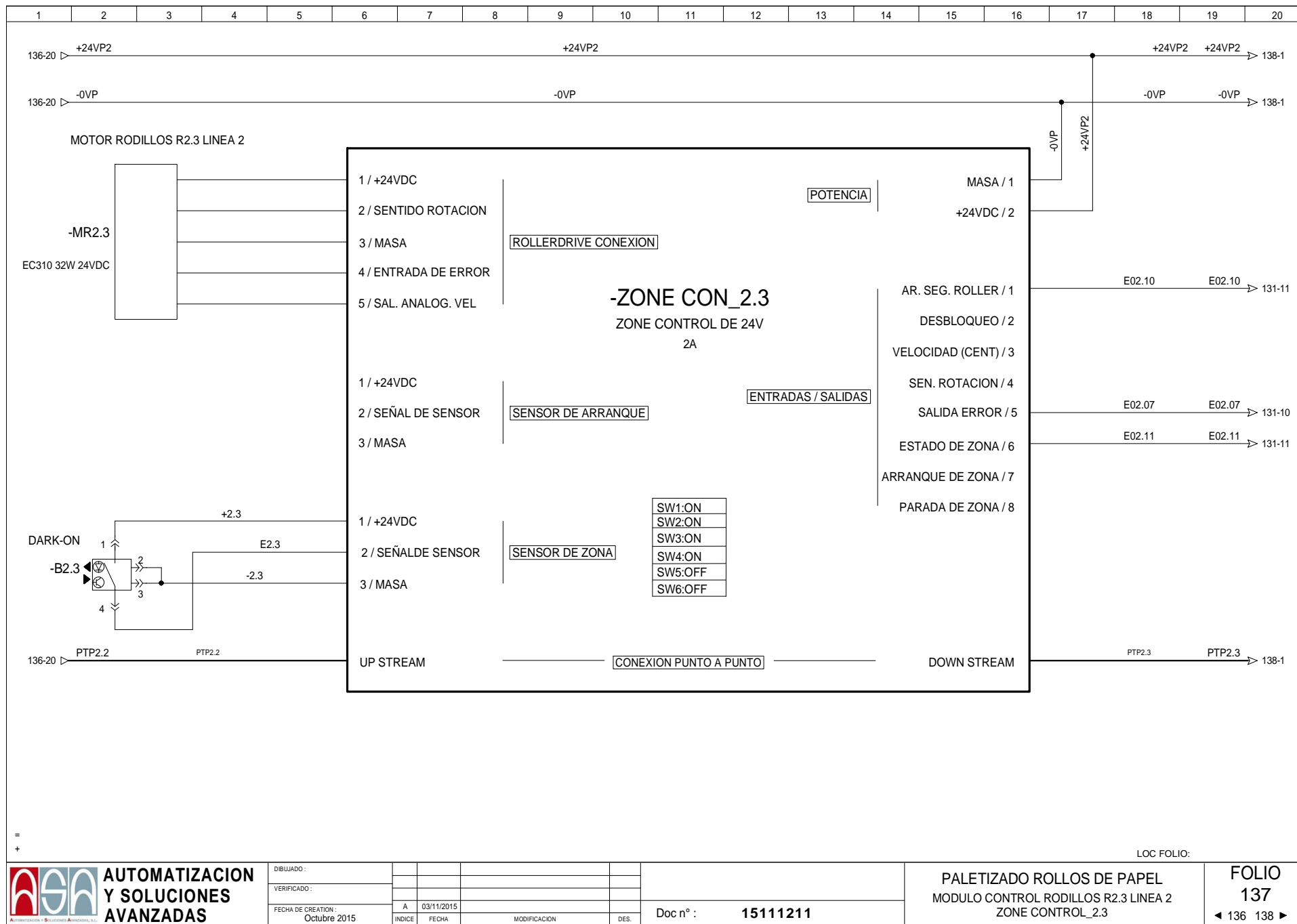
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
MODULO CONTROL RODILLOS R2.2 LINEA 2
ZONE CONTROL_2.2

LOC FOLIO:
FOLIO 136
◀ 135 137 ▶



138-1

138-1

PTP2.3



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

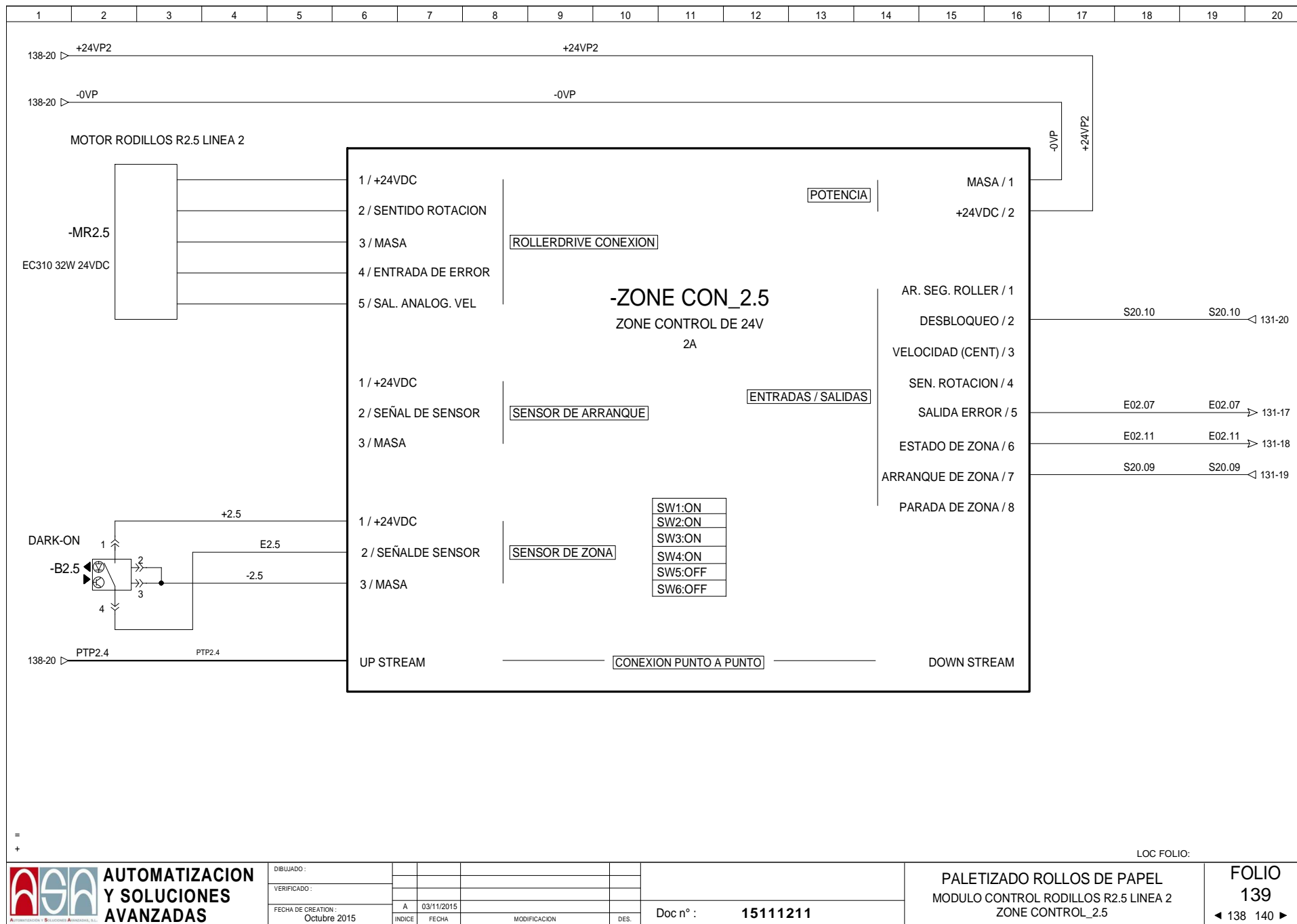
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
MODULO CONTROL RODILLOS R2.3 LINEA 2
ZONE CONTROL_2.3

LOC FOLIO:

FOLIO
137

◀ 136 138 ▶

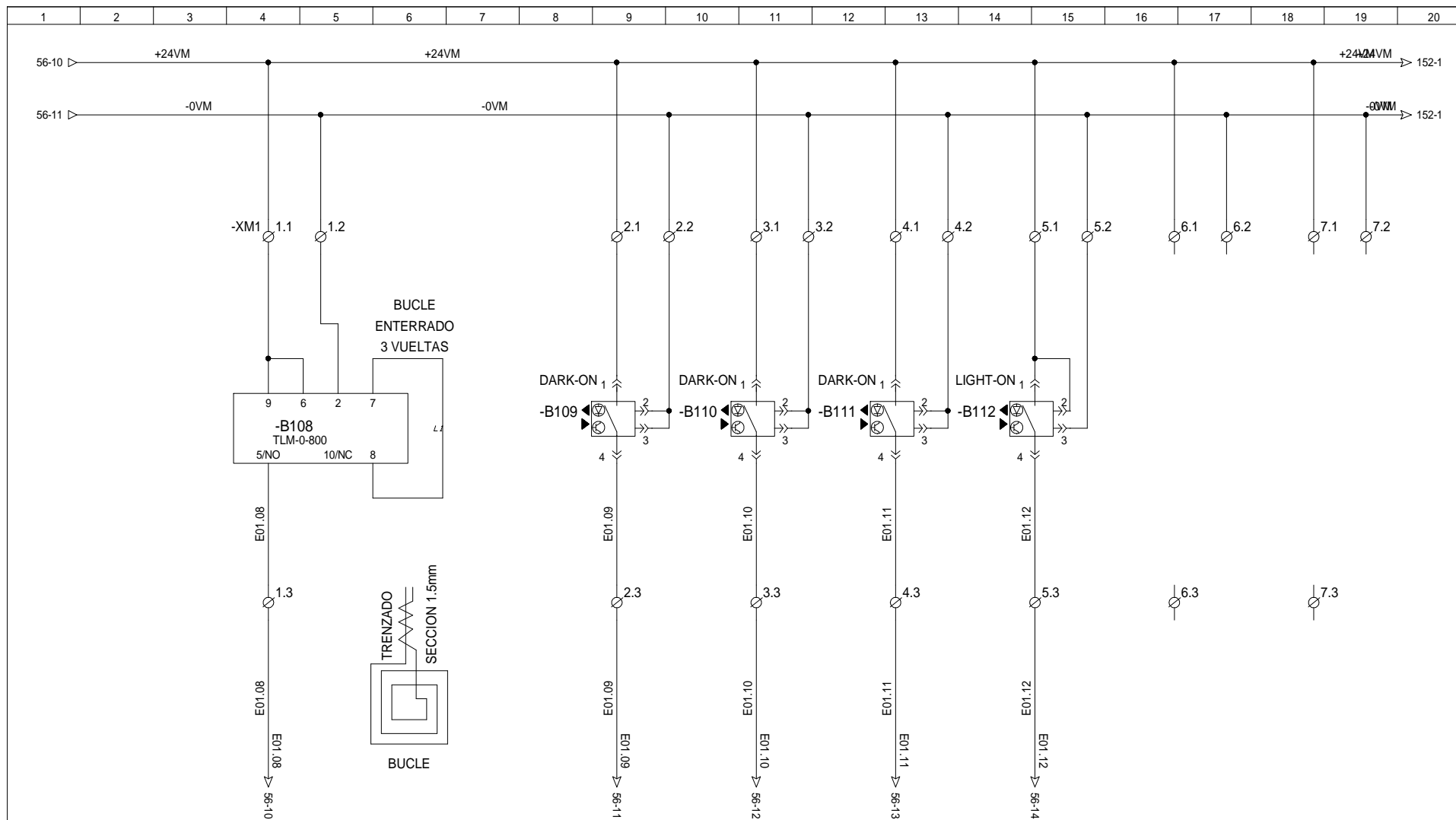


=

+

LOC FOLIO:





Nota: La longitud del cable de enlace deberá ser la mínima posible y no sobrepasar los 200m. Si esta distancia es superior la inductancia que presenta el cable de enlace disminuye la sensibilidad.

Los 2 conductores deberán estar trenzados a razón de unos 10 a 20 cruces por metro de longitud: se debe poner atención en evitar la proximidad de otros cables eléctricos que alimenten motores,

+ C3

LOC FOLIO: C3



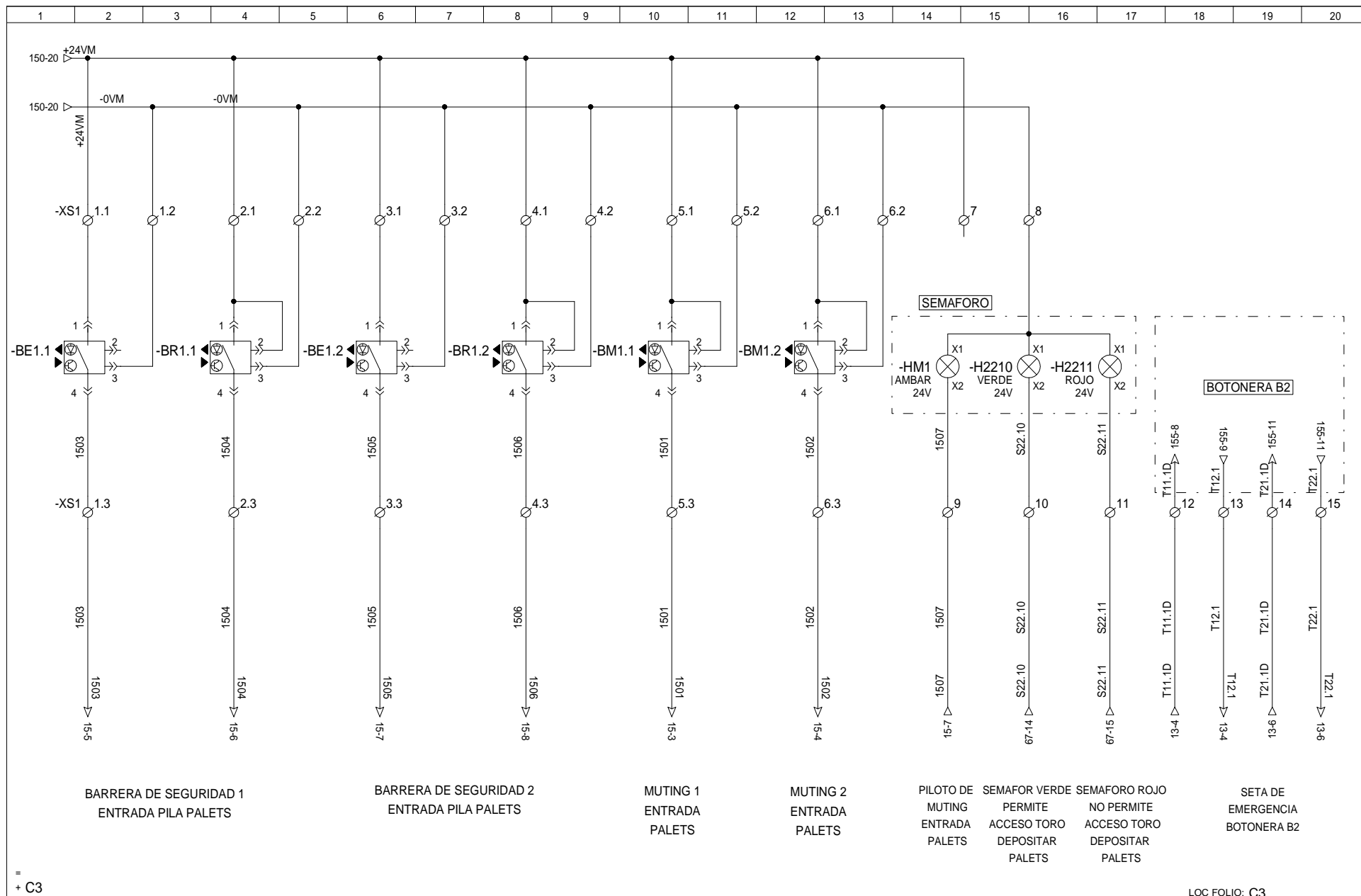
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

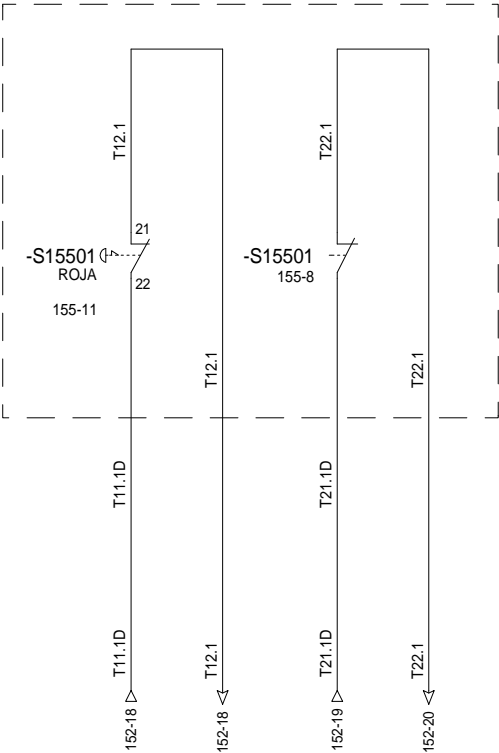
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 3
ALIMENTADOR DE PALETS

FOLIO
150
◀ 140 152 ▶



BOTONERA 1 ELEMENTO



=
+ B2

LOC FOLIO: B2



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

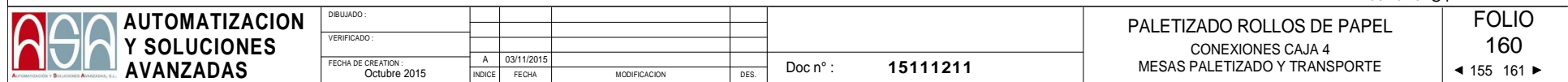
DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
INDICE	FECHA	MODIFICACION	DES.	

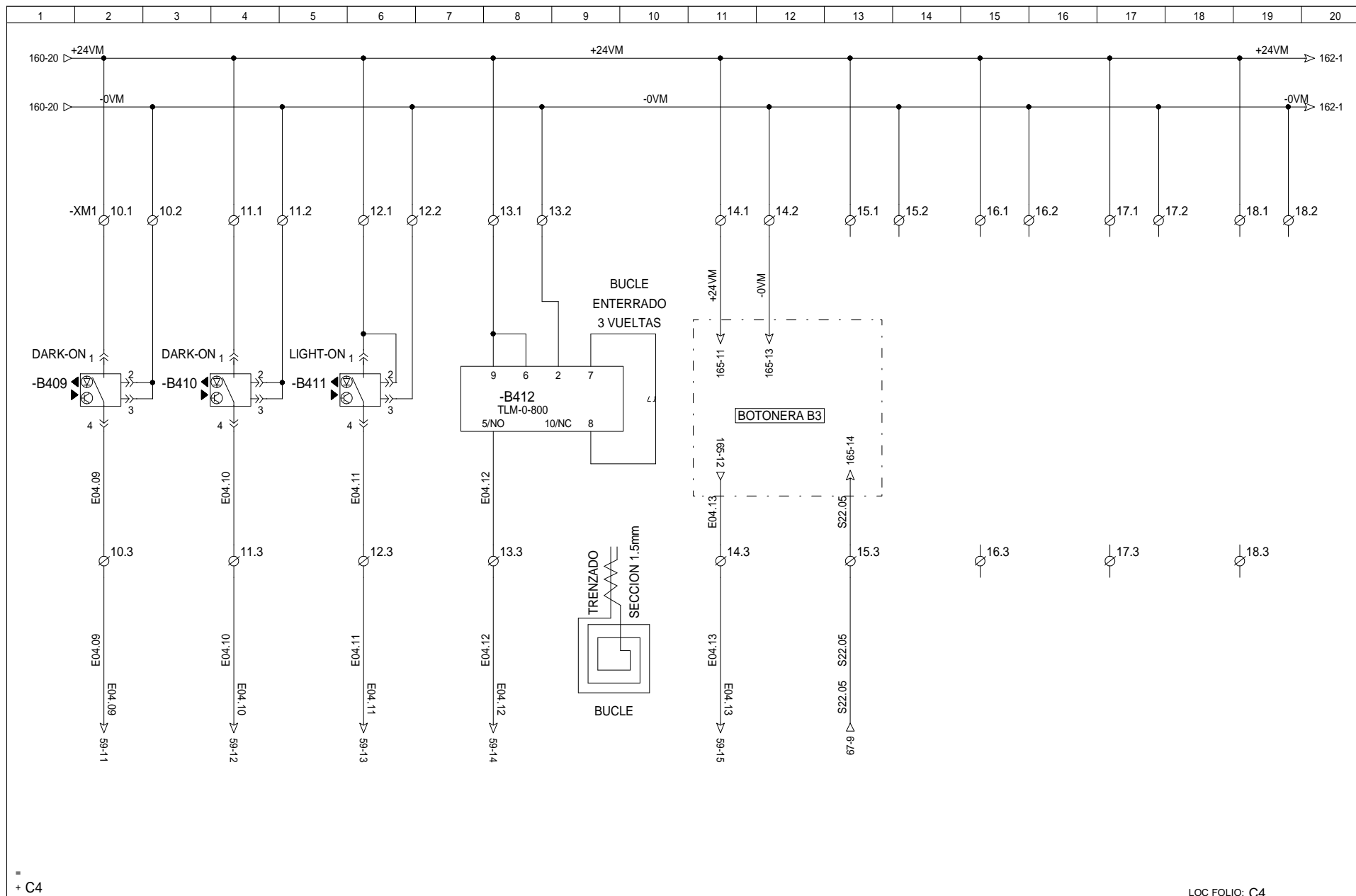
Doc n° : 15111211

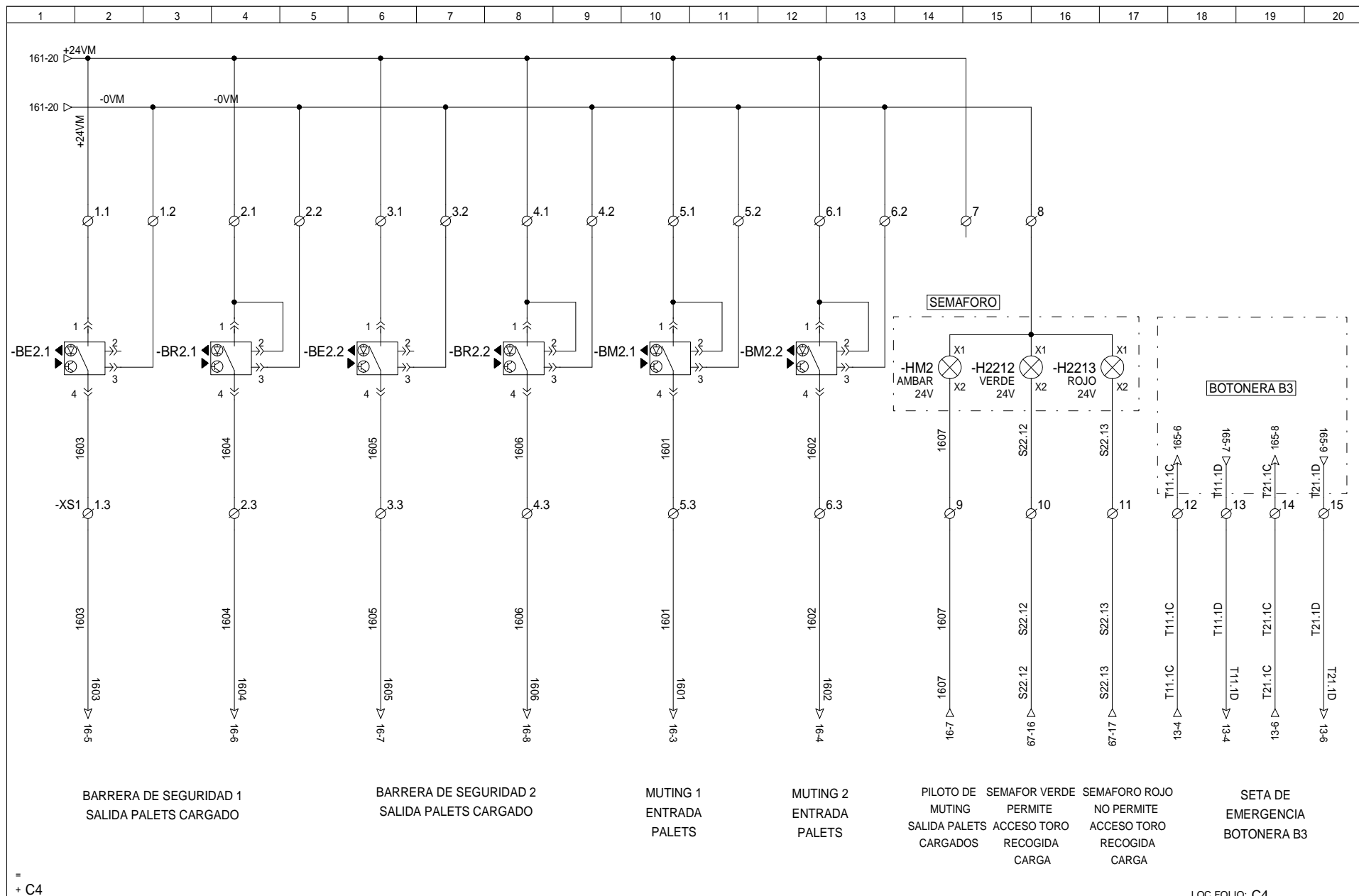
PALETIZADO ROLLOS DE PAPEL
BOTONERA DEPOSITA MONTACARGAS (B2)
CONEXIONES

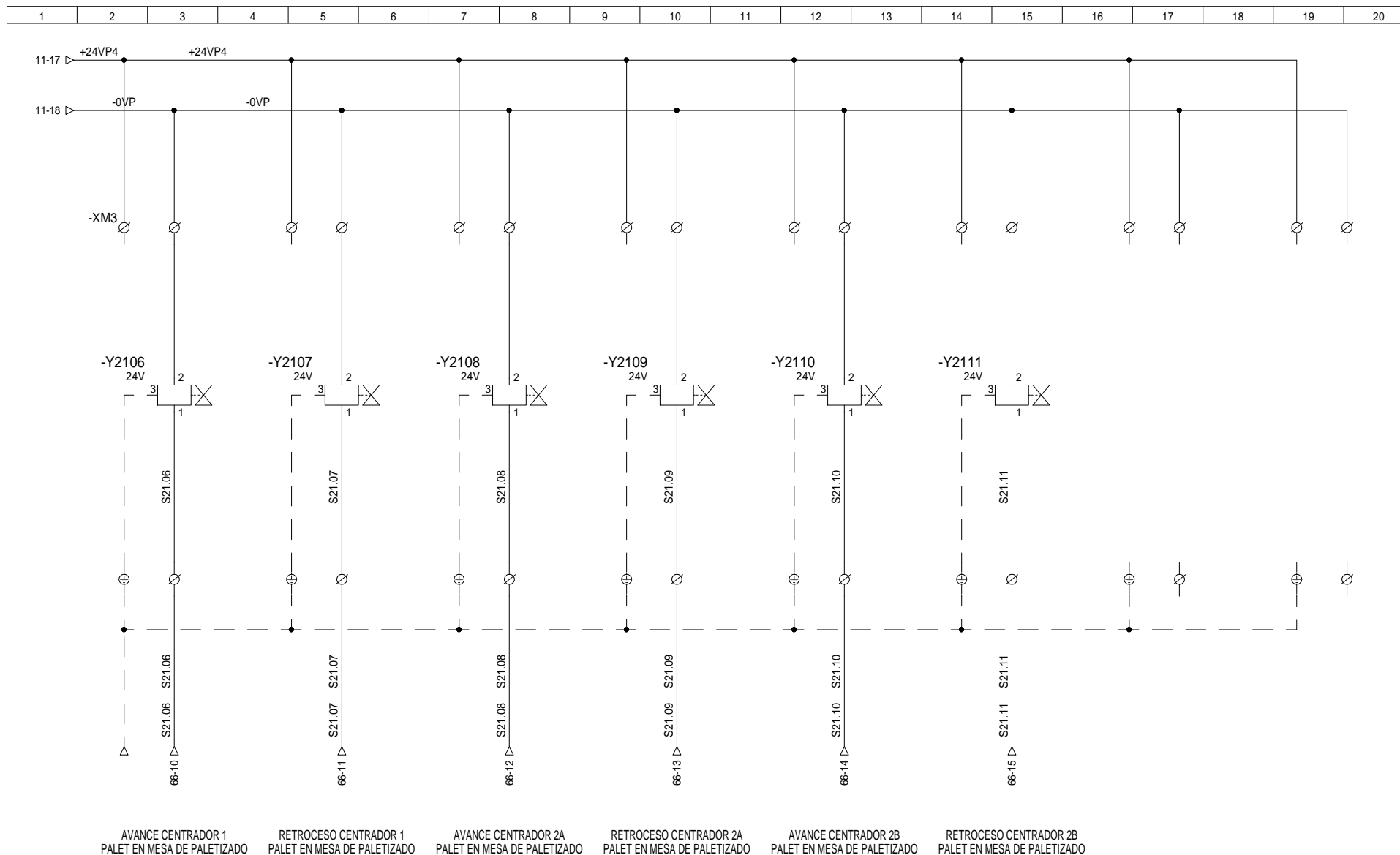
FOLIO
155

◀ 152 160 ▶









=
+ C4

LOC FOLIO: C4



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
INDICE	FECHA	MODIFICACION	DES.	

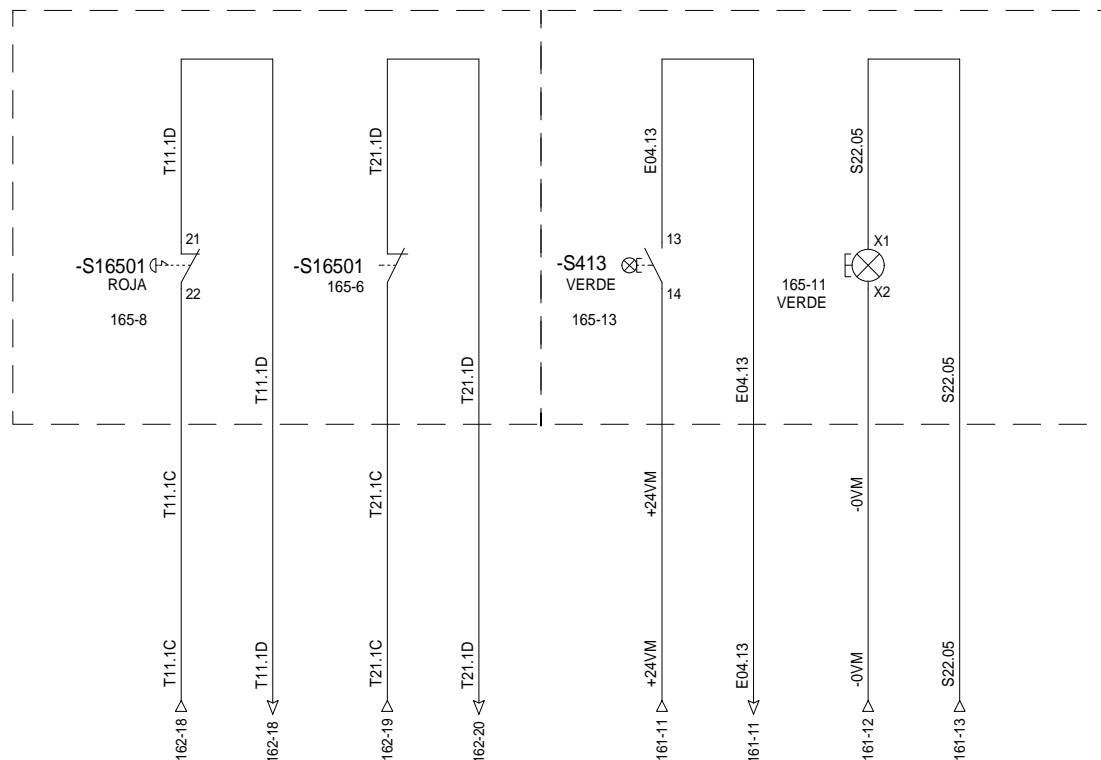
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONES CAJA 4
MESAS PALETIZADO Y TRANSPORTE

**FOLIO
163**

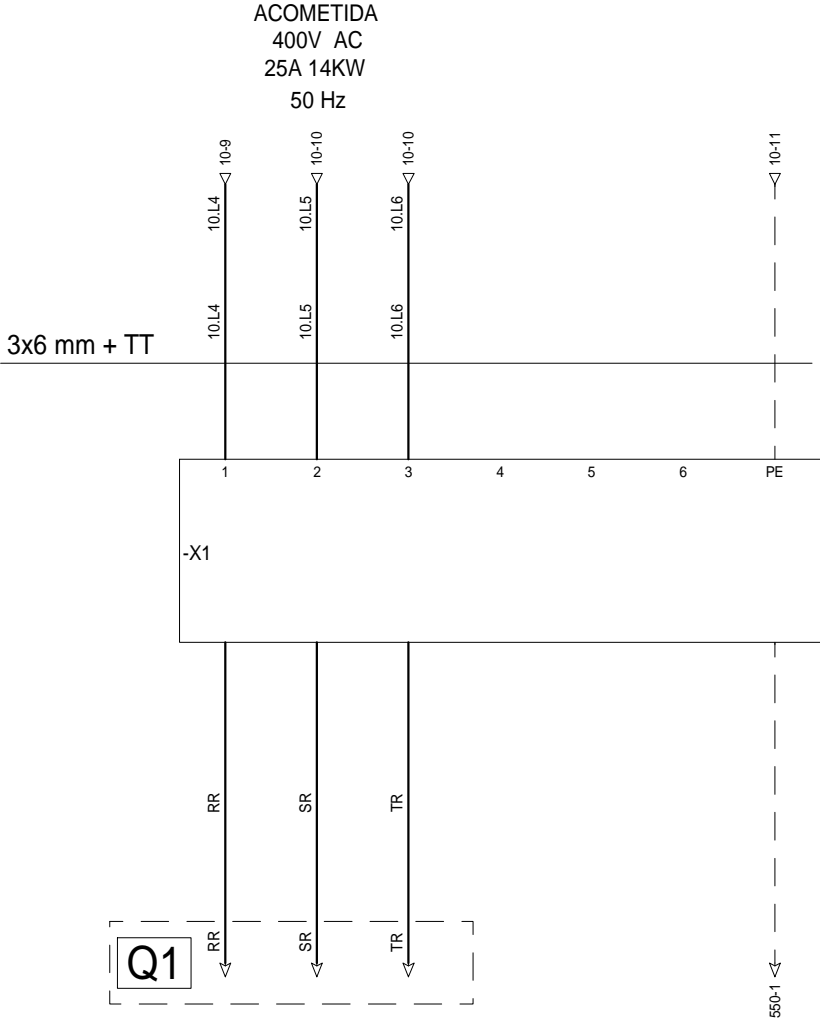
◀ 162 165 ▶

BOTONERA 2 ELEMENTOS

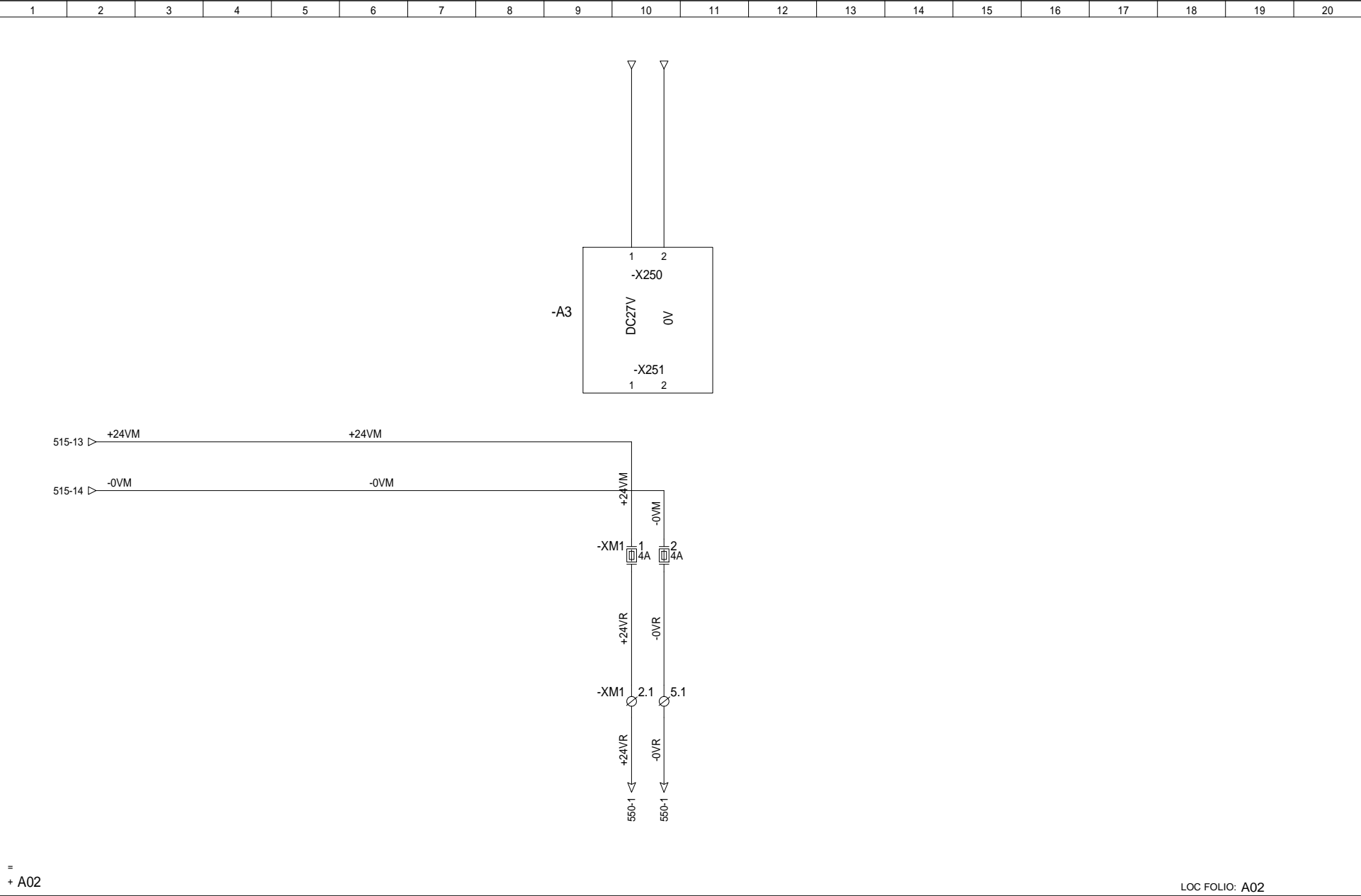


=
+ B3

LOC FOLIO: B3



=
+ A02



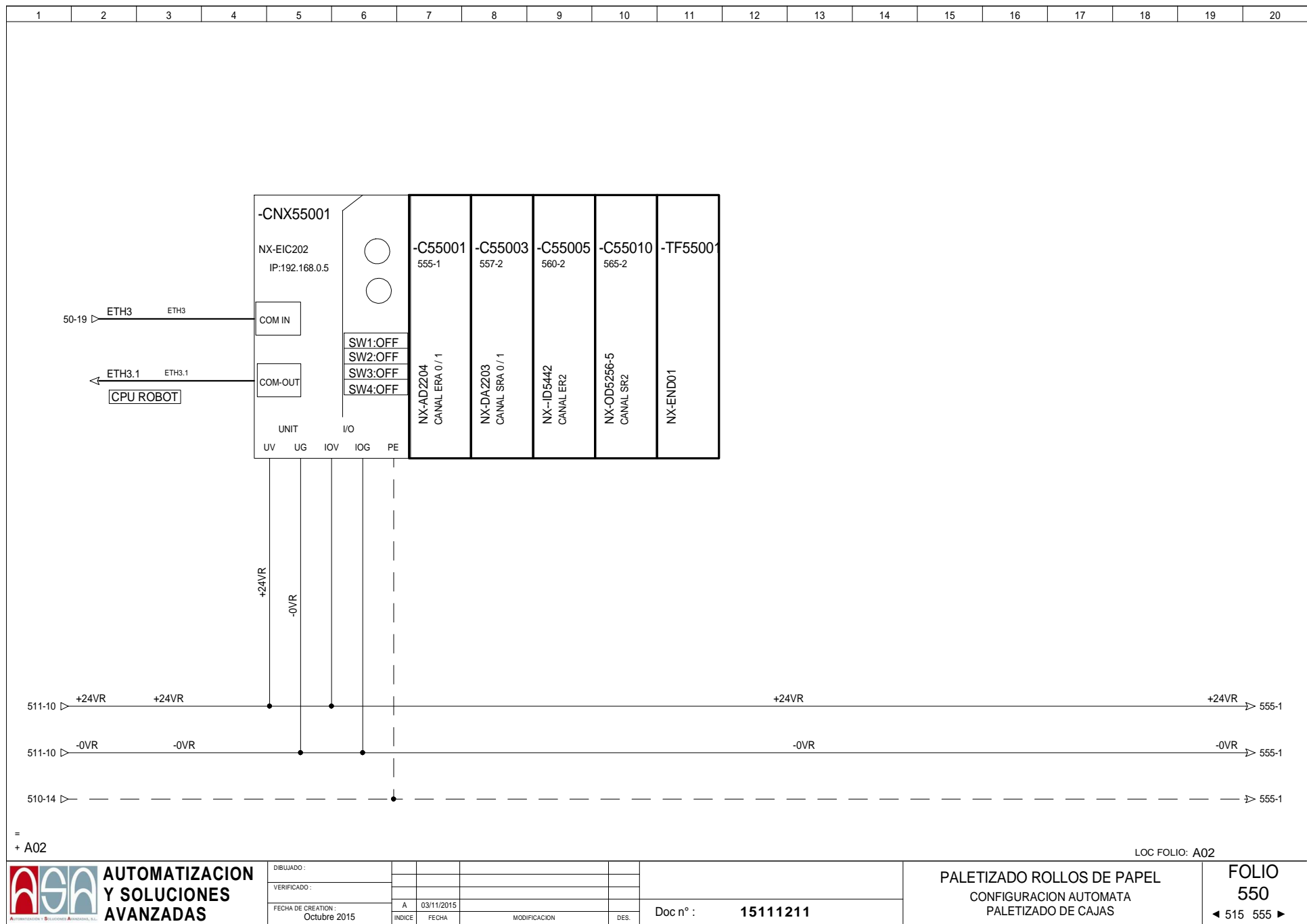
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

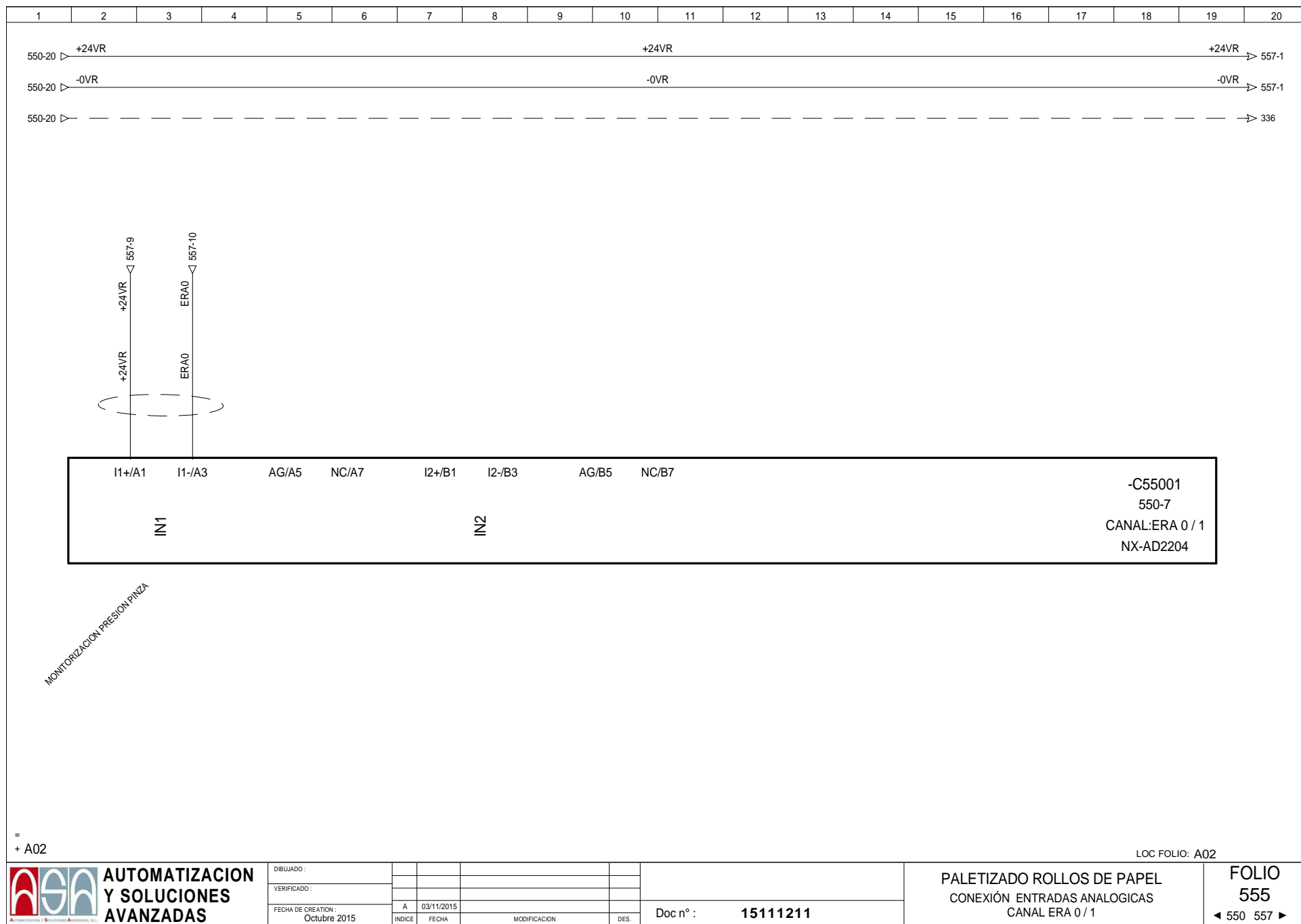
DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
INDICE	FECHA	MODIFICACION	DES.	

Doc n° : **15111211**

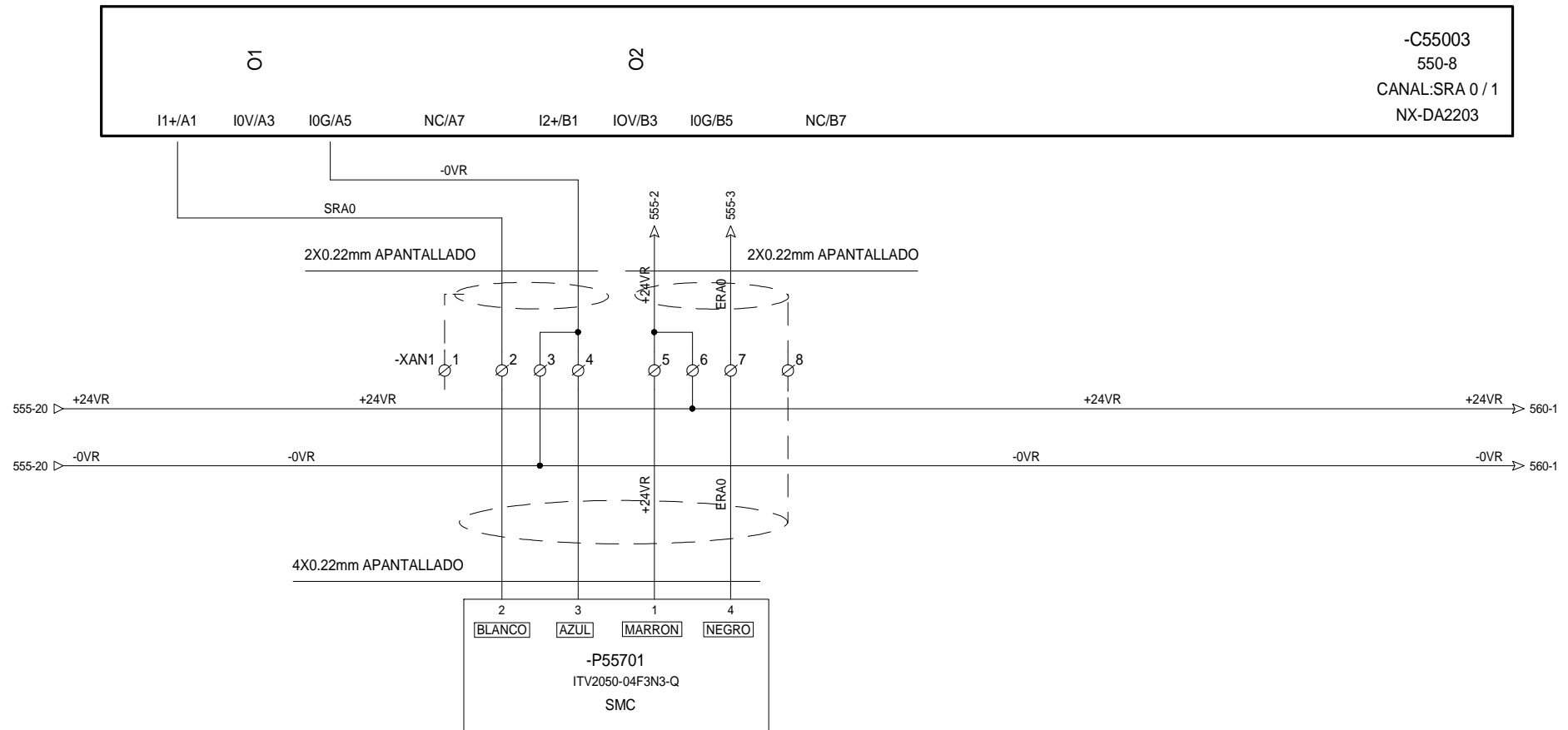
PALETIZADO ROLLOS DE PAPEL
DISTRIBUCION

LOC FOLIO: A02
**FOLIO
511**
◀ 510 515 ▶





1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----


$$+ A02$$

LOC FOLIO: A02



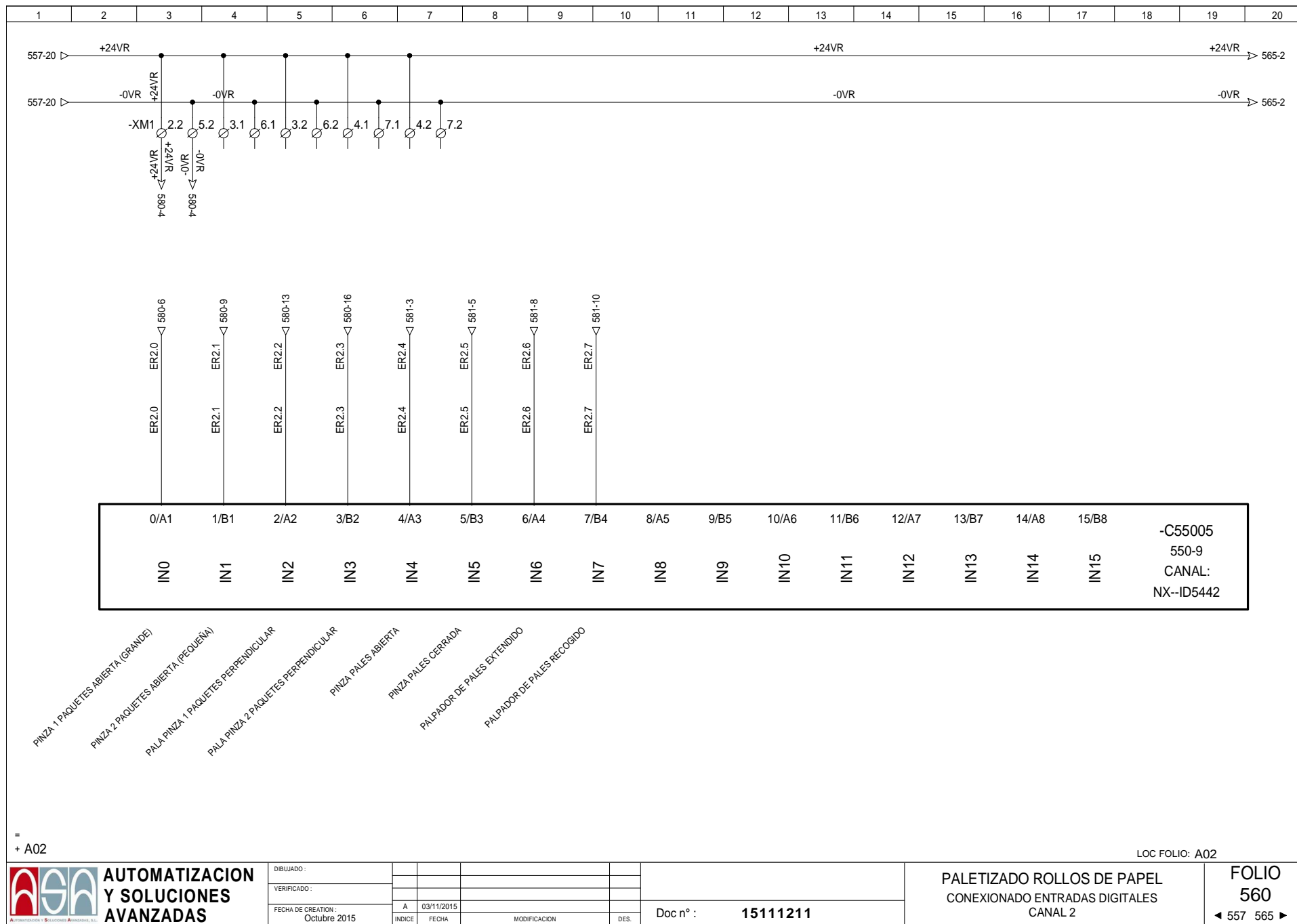
AUTOMATIZACION Y SOLUCIONES AVANZADAS

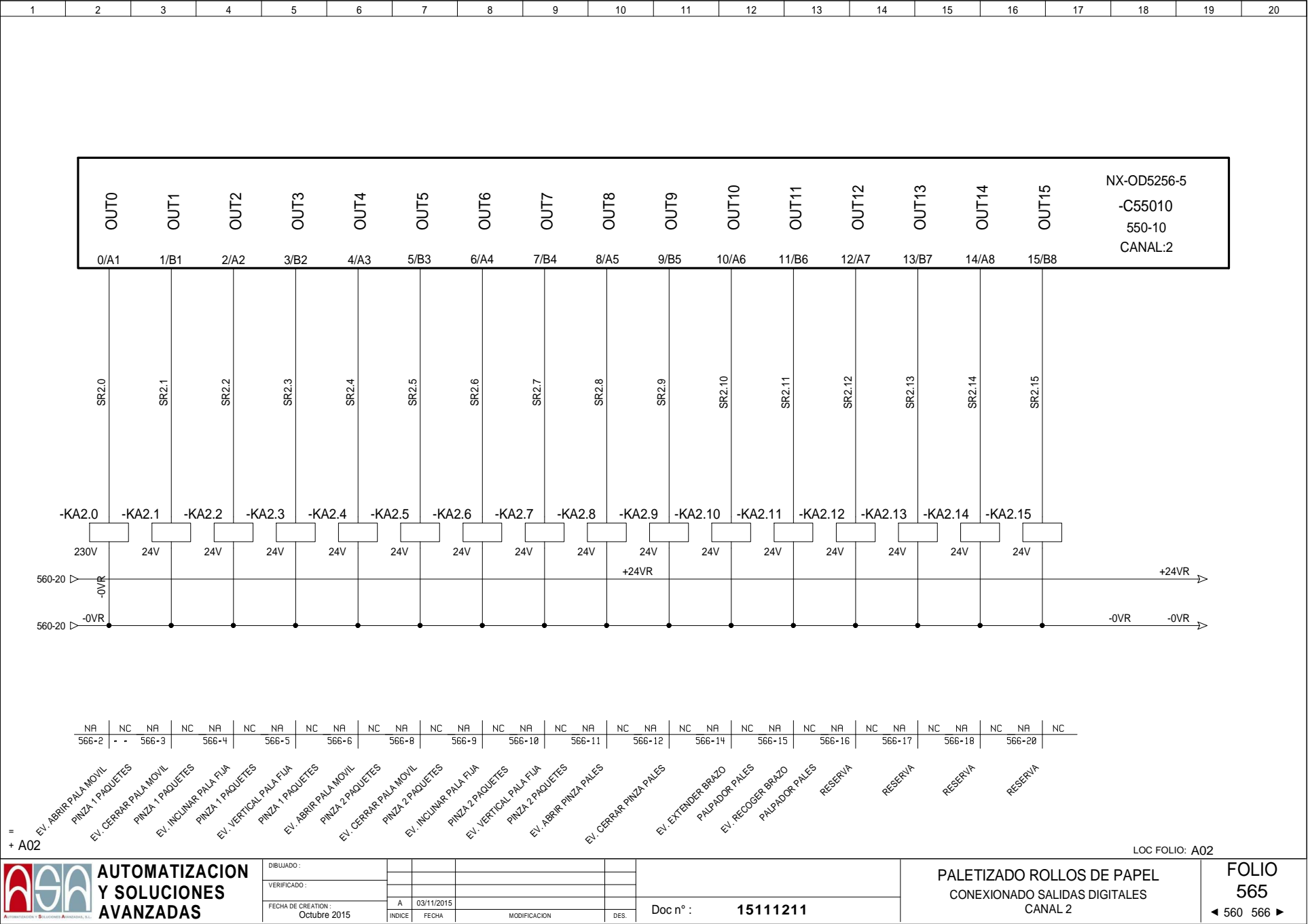
DIBUJADO :				
VERIFICADO :				
FECHA DE CREATION :	A	03/11/2015		
Octobre 2015	INDICE	FECHA	MODIFICACION	DES.

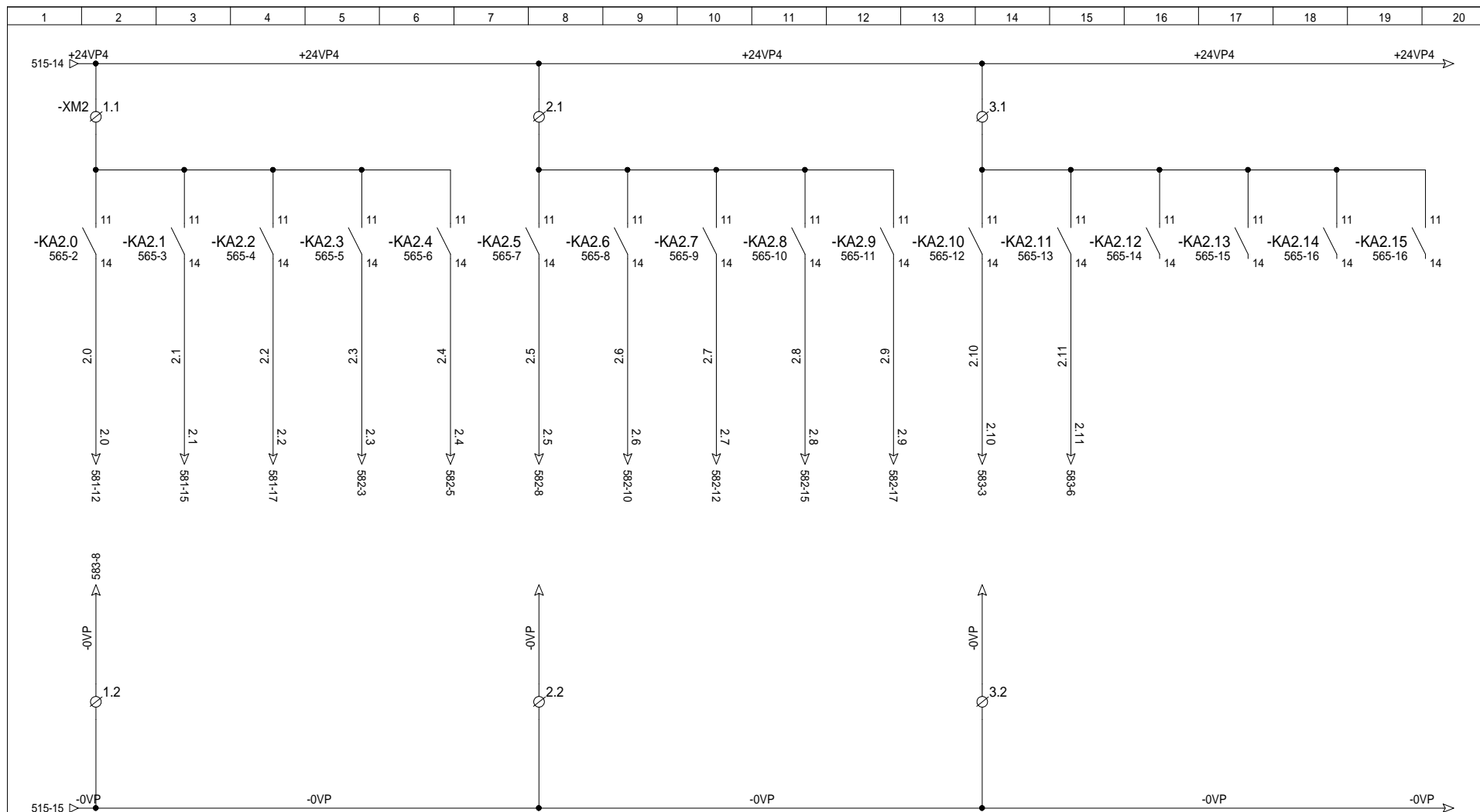
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIÓN SALIDAS ANALÓGICAS
CANAL SRA 0 / 1

FOLIO
557







=
+ A02

LOC FOLIO: A02



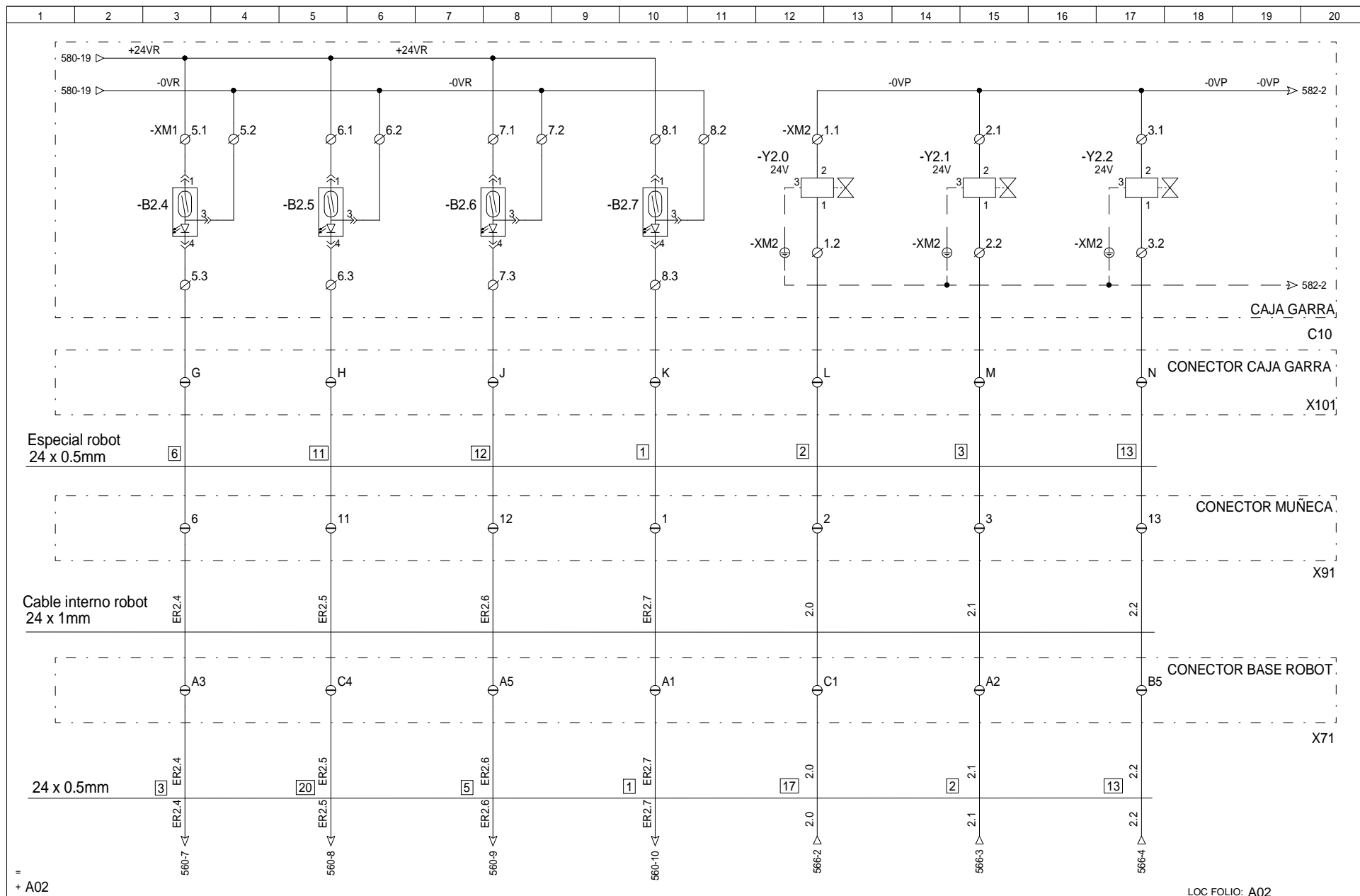
**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

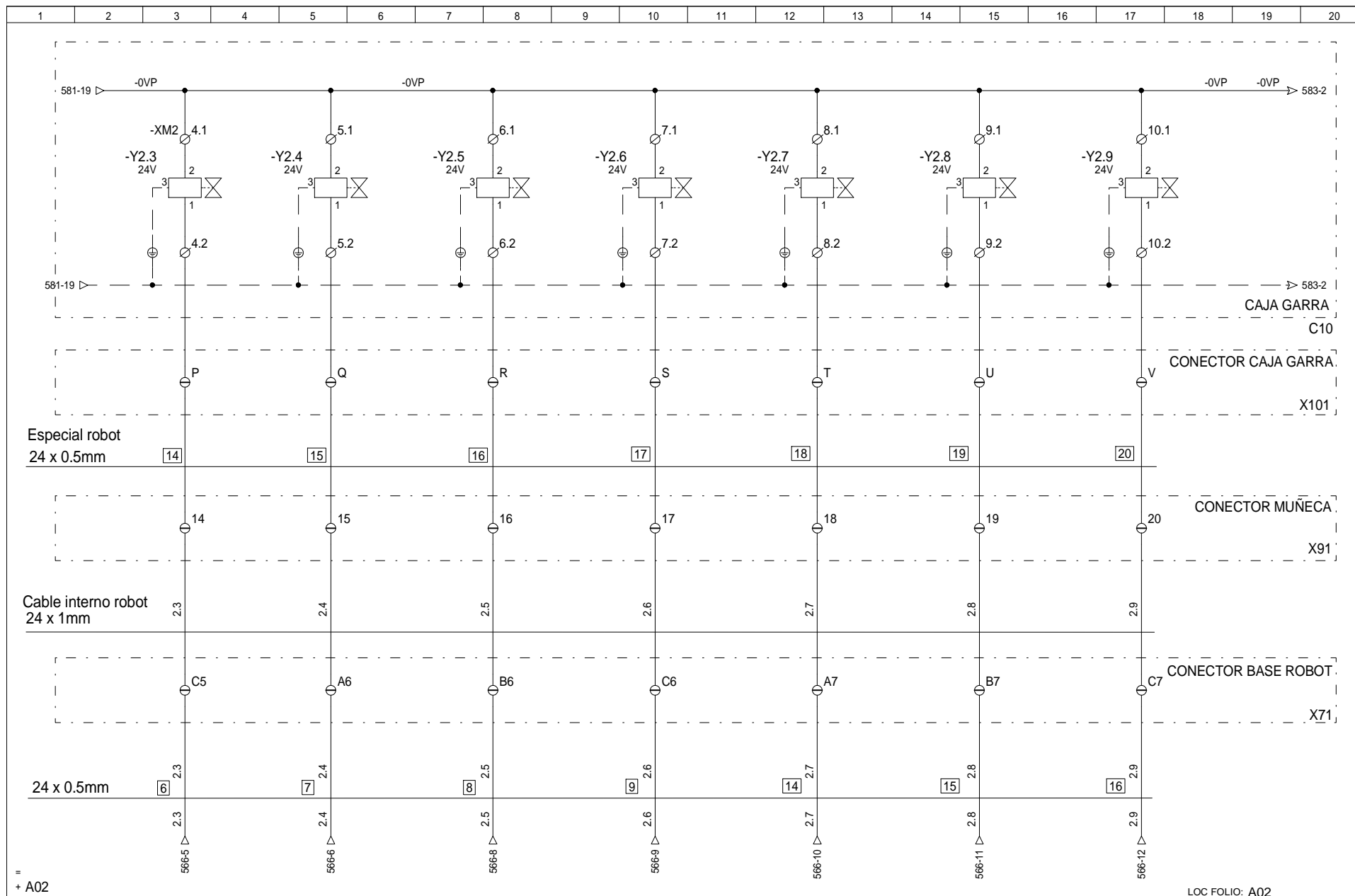
DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
Octubre 2015	INDICE	FECHA	MODIFICACION	DES.

Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
CONEXIONADO SALIDAS DIGITALES
CANAL 2

FOLIO
566
◀ 565 580 ▶





**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS**

DIBUJADO:

VERIFICADO:

FECHA DE CREATION:
Octubre 2015

INDICE

FECHA

MODIFICACION

DES.

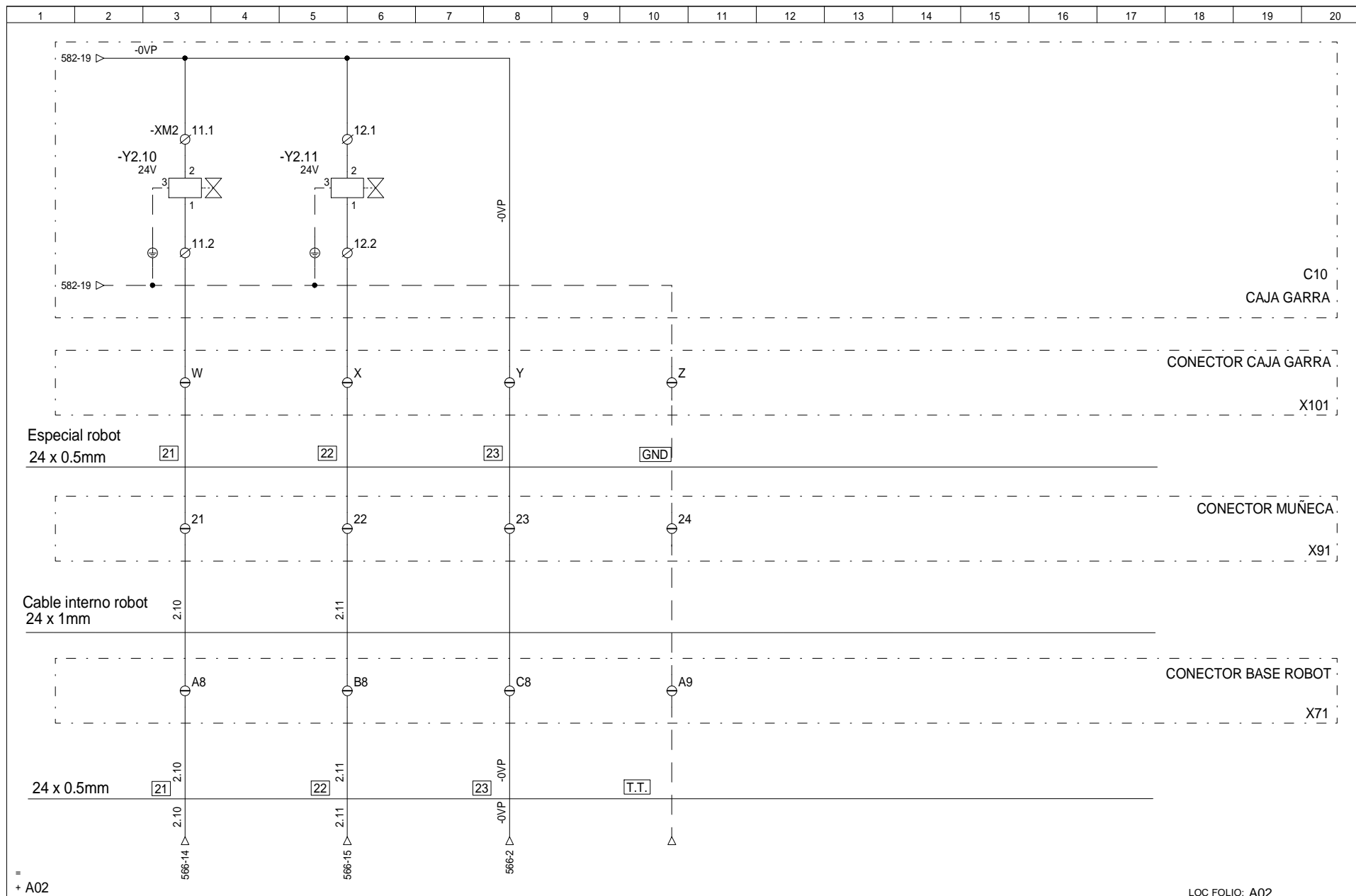
Doc n°:

15111211

PALETIZADO ROLLOS DE PAPEL
ROBOT
CONEXION GARRA

**FOLIO
582**

◀ 581 583 ▶



A		03/11/2015			
INDICE	MODIFICACION	FECHA	DIBUJADO	VERIFICADO	APPROBADO

AUTOMATIZACION Y SOLUCIONES AVANZADAS S.L. Av. de la Estacion, 12 - Pol. Ind. "LAS CANTERAS" 45520 VILLALUENGA DE LA SAGRA (TOLEDO) TF +34 925537970 info@asasl.es FAX: +34 925537954	DIBUJADO:
	VERIFICADO:
	FECHA DE CREACION: Octubre 2015
	PROYECTO N°: 15111211

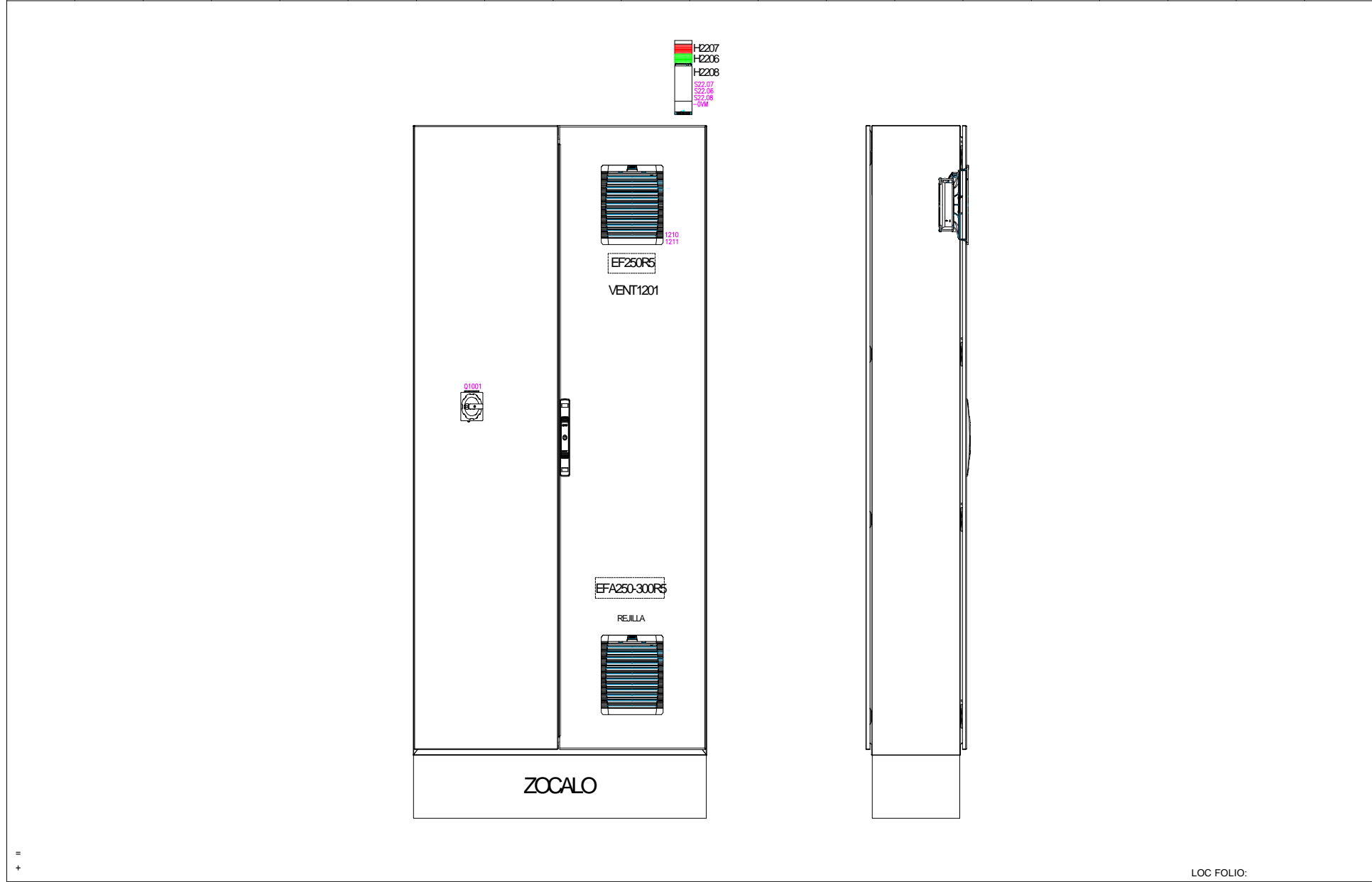


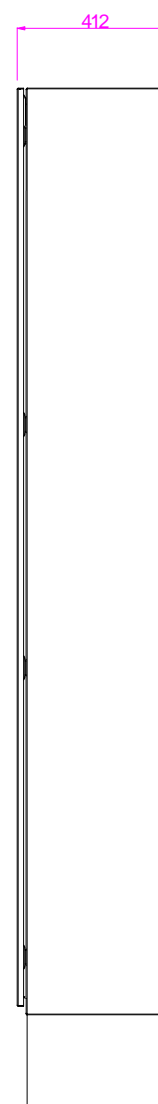
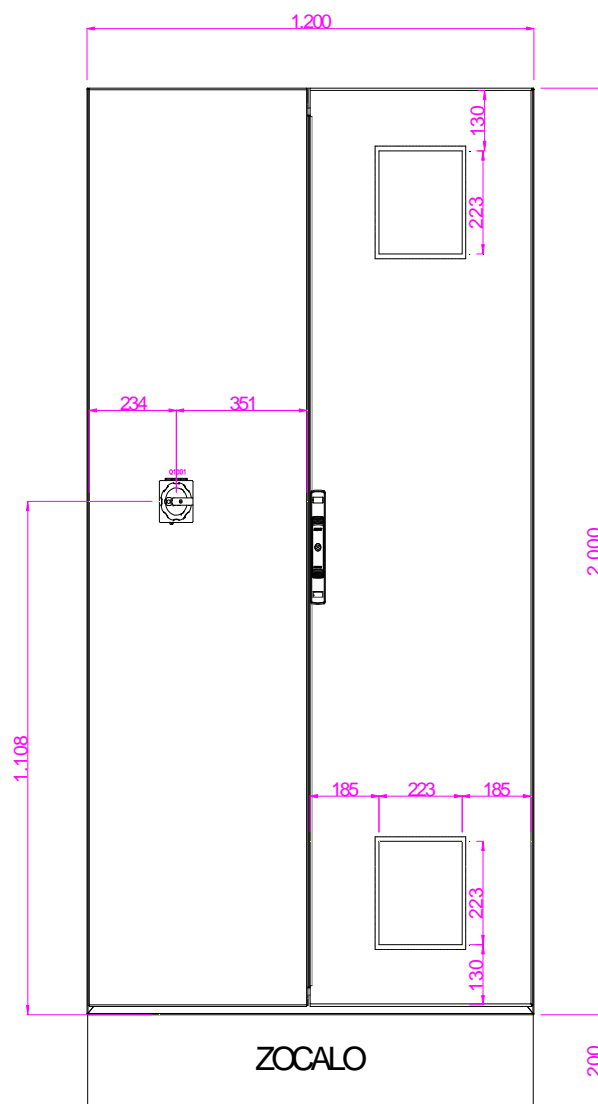
PALETIZADO ROLLOS DE PAPEL CON ROBOT MECANIZADOS PORTADA

DOCUMENTO N° : 15111211	1 / 12
--------------------------------	---------------

ESTE DOCUMENTO ES PROPIEDAD DE AUTOMATIZACION Y SOLUCIONES AVANZADAS.
ESTA PROHIBIDA TODA REPRODUCCION TOTAL O PARCIAL SIN AUTORIZACION.

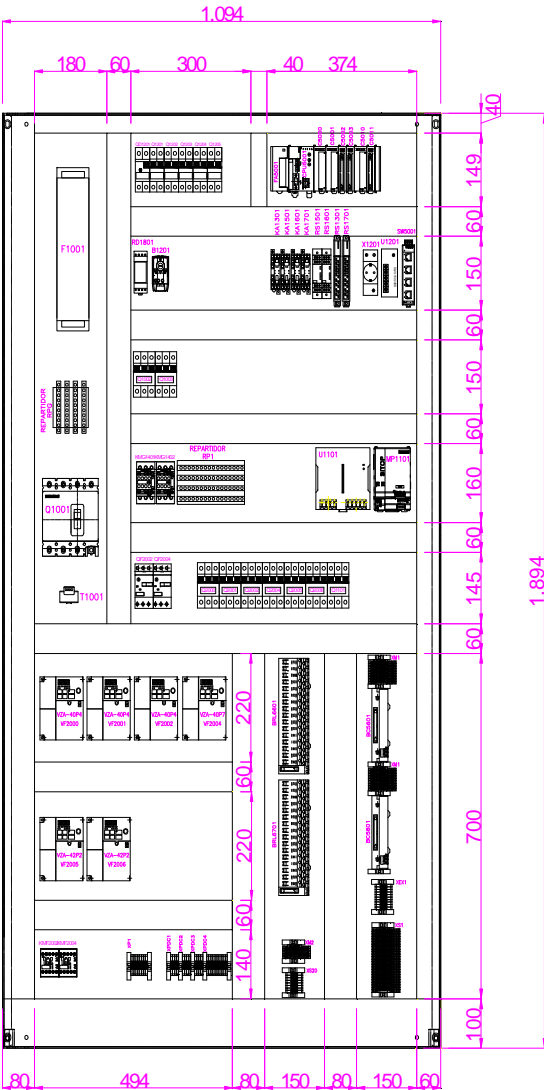
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

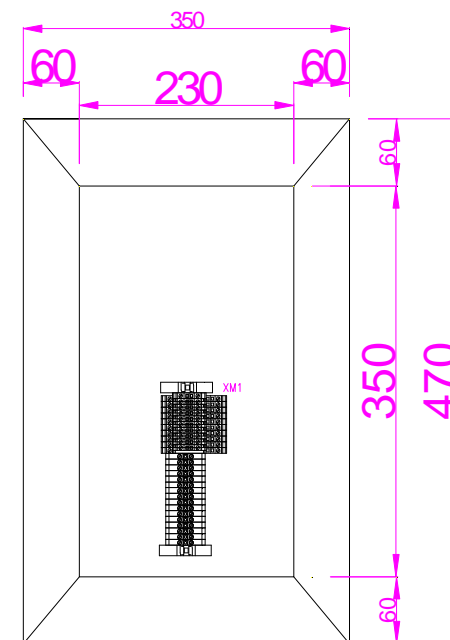
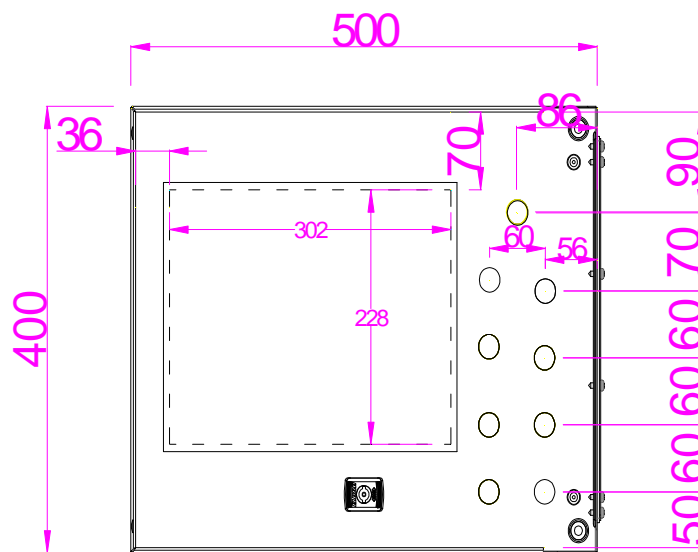
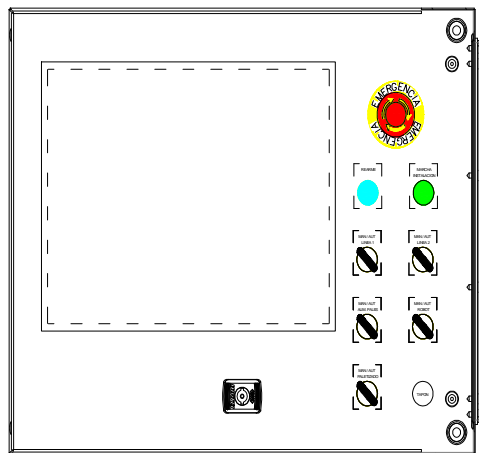




NOTAS:
LAS COTAS DEL SECCIONADOR SON
ORIENTATIVAS.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----





VISTA INTERIOR

ETIQUETAS

S10001

T11.1B / T11.1C

T21.1B / T21.1C

S106

S10002

+24VM / E01.06

T31.1 / T31.1A

S22.06 -0VM

T31.7 / T31.7A

1703 -0VM

S101

S100

+24VM / E01.01

+24VM / E01.00

S103

S102

+24VM / E01.03

+24VM / E01.02

E01.04

+24VM / T31.2A



**AUTOMATIZACION
Y SOLUCIONES
AVANZADAS S.L.**

DIBUJADO:				
VERIFICADO:				
FECHA DE CREATION:	A	03/11/2015		
INDICE	FECHA	MODIFICACION	DES.	

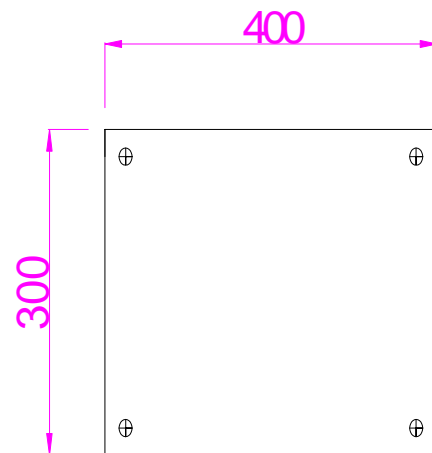
Doc n° : **15111211**

PALETIZADO ROLLOS DE PAPEL
PUPITRE
P1

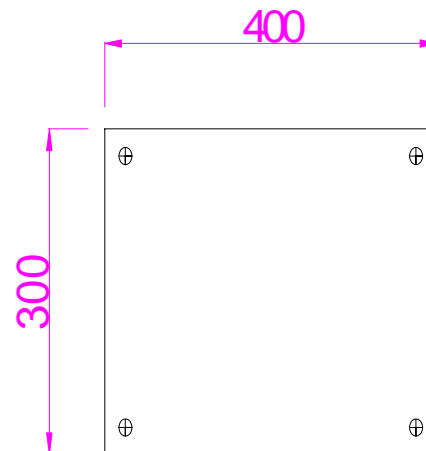
FOLIO
11

◀ 10 15 ▶

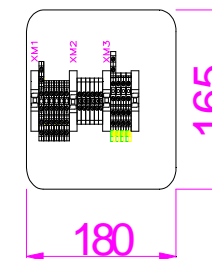
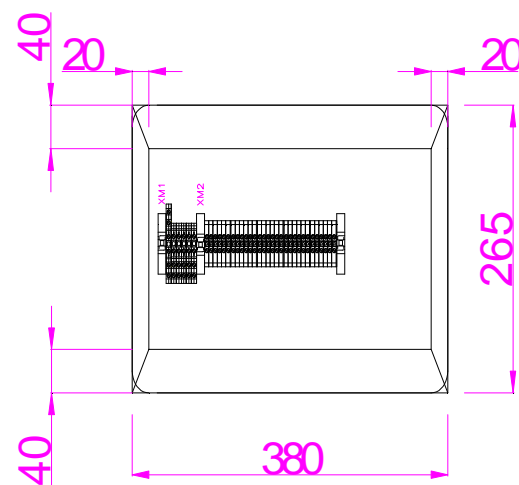
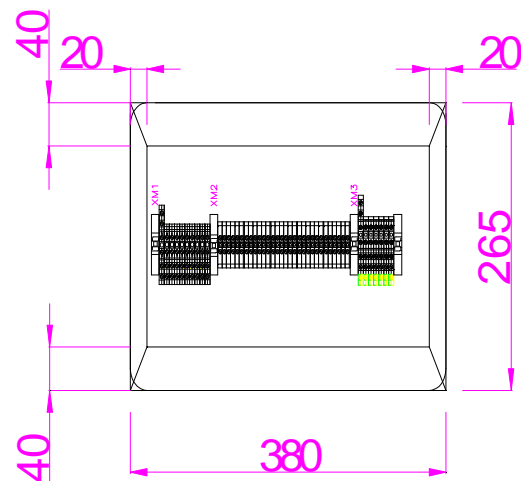
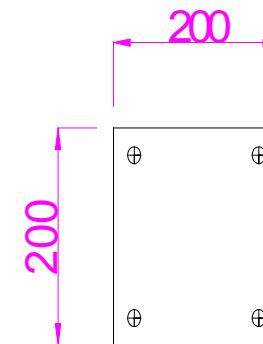
CAJA C1
LINEA DE TRANSPORTE 1



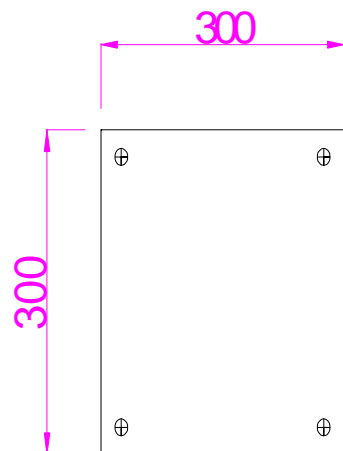
CAJA C2
LINEA DE TRANSPORTE 2



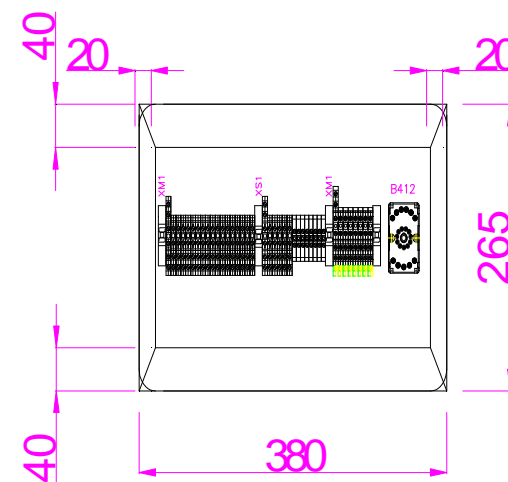
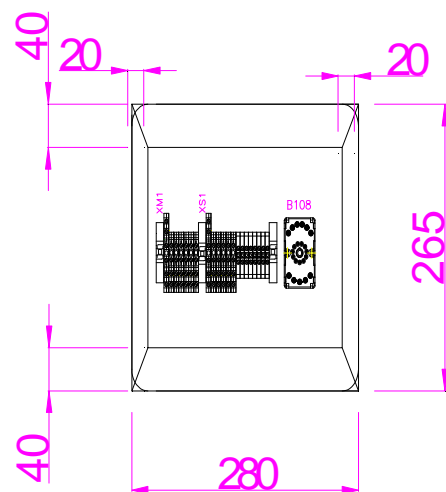
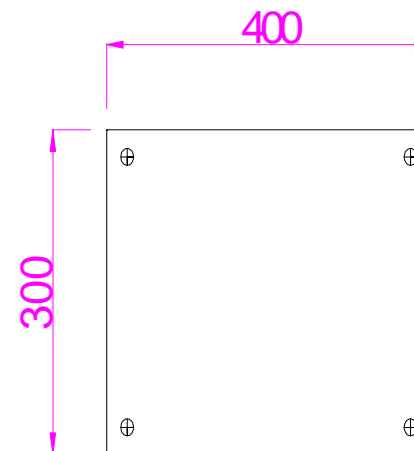
CAJA C2.1
LINEA DE TRANSPORTE 2

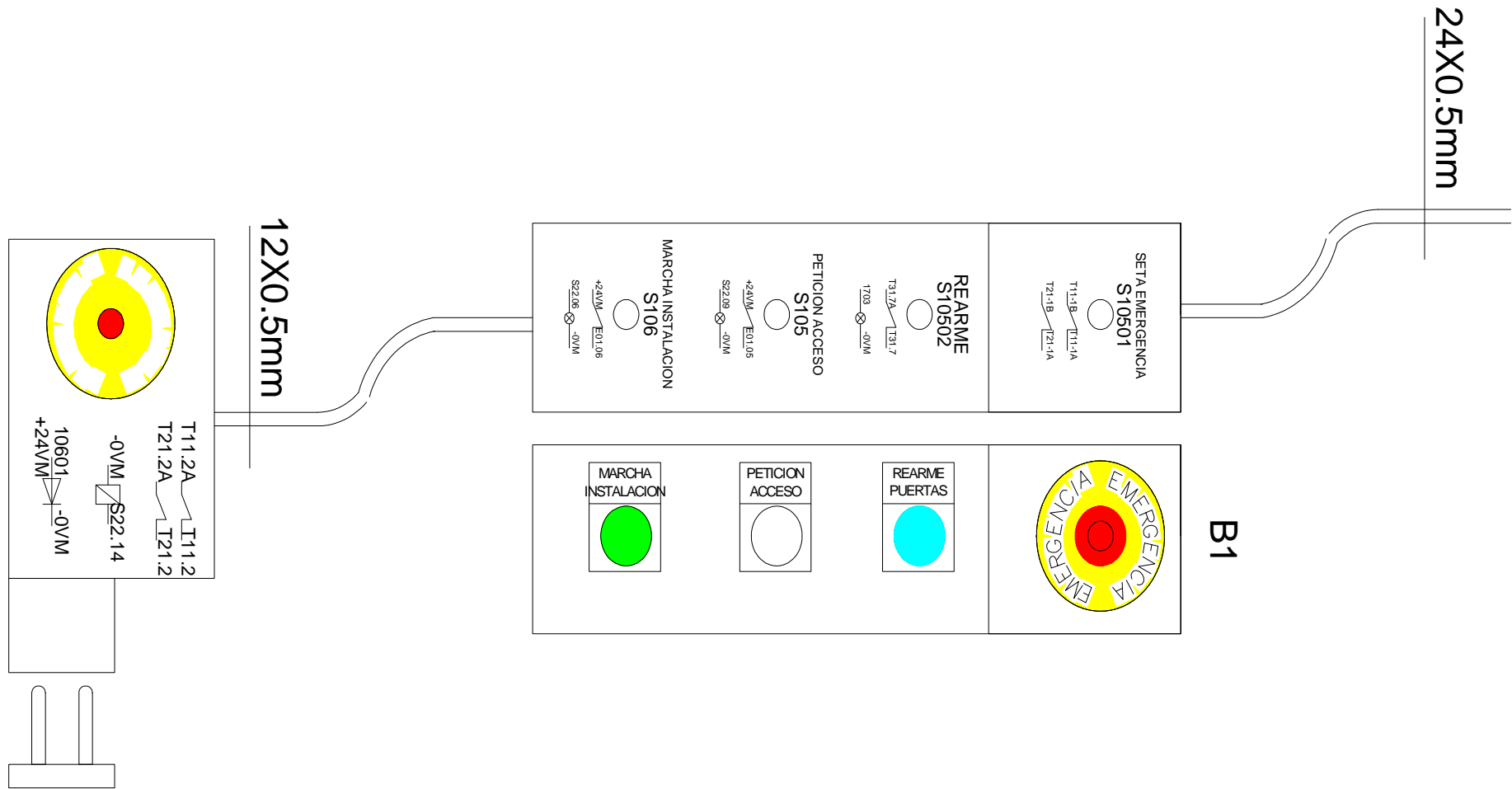


CAJA C3
ALIMENTADOR DE PALETS



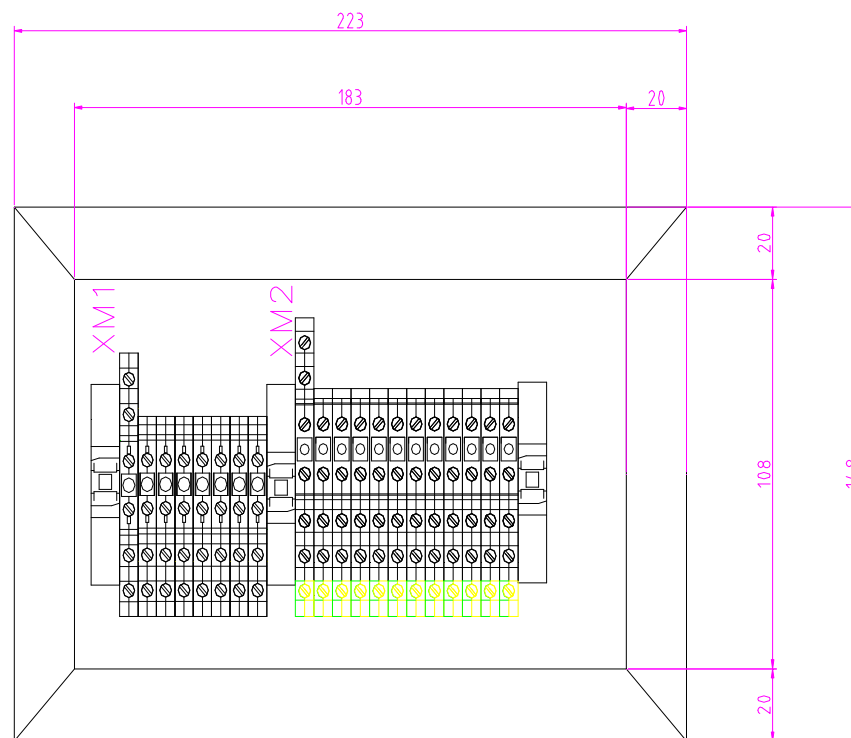
CAJA C4
LINEA DE PALETIZADO





CAJA C10

GARRA ROBOT



=
+

B	COPIA TALLER	03/11/2015	FPR		
INDICE	MODIFICACION	FECHA	DIBUJADO	VERIFICADO	APPROBADO

AUTOMATIZACION Y SOLUCIONES AVANZADAS S.L. Av. de la Estacion, 12 - Pol. Ind. "LAS CANTERAS" 45520 VILLALUENGA DE LA SAGRA (TOLEDO) TF +34 925537970 info@asasl.es FAX: +34 925537954	DIBUJADO:
	VERIFICADO:
	FECHA DE CREACION: Octubre 2015
	PROYECTO N°: 15111211

PALETIZADO ROLLOS DE PAPEL

CON ROBOT

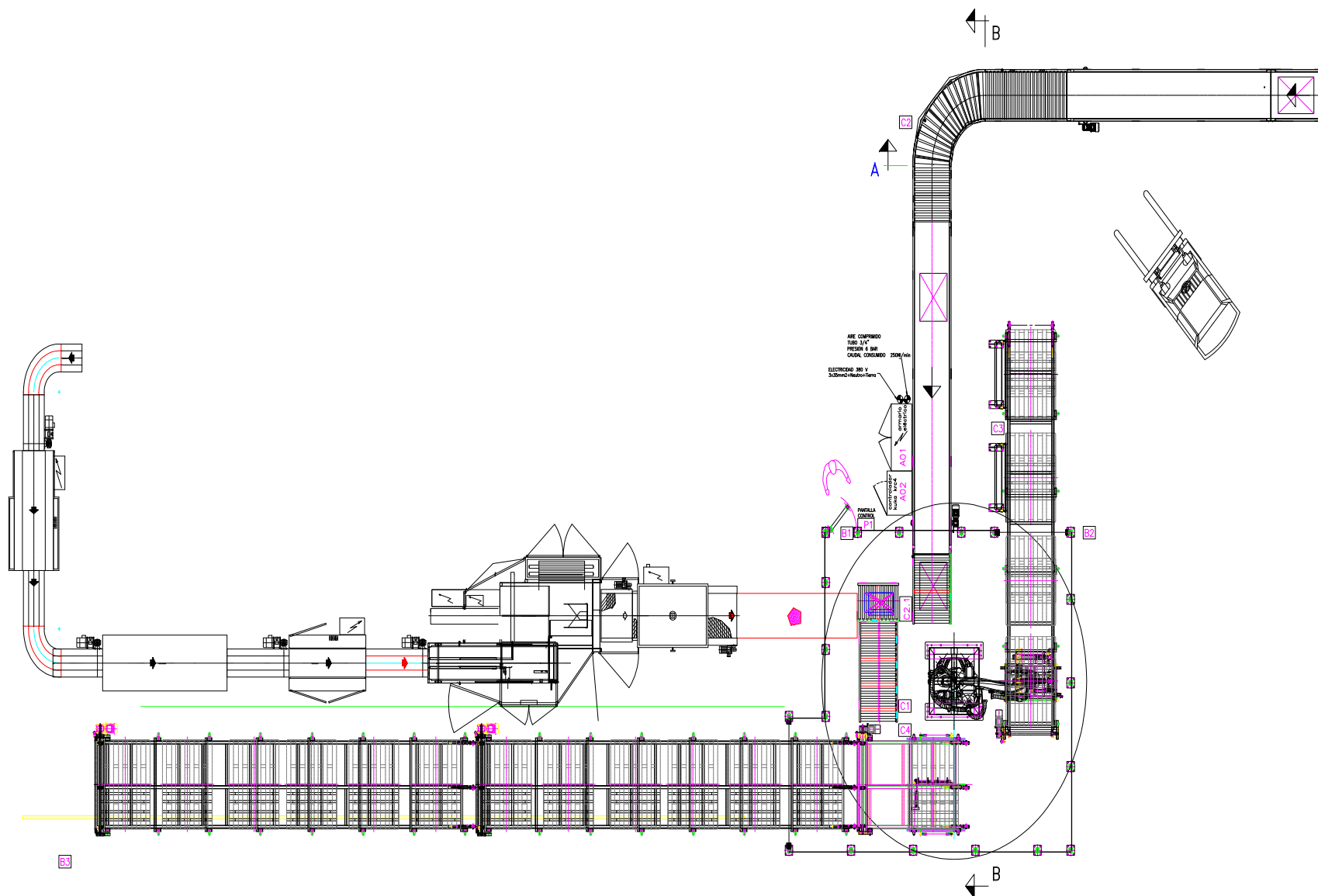
MAQUINAS


PORTADA

DOCUMENTO N° : 15111211	1 / 9
--------------------------------	--------------

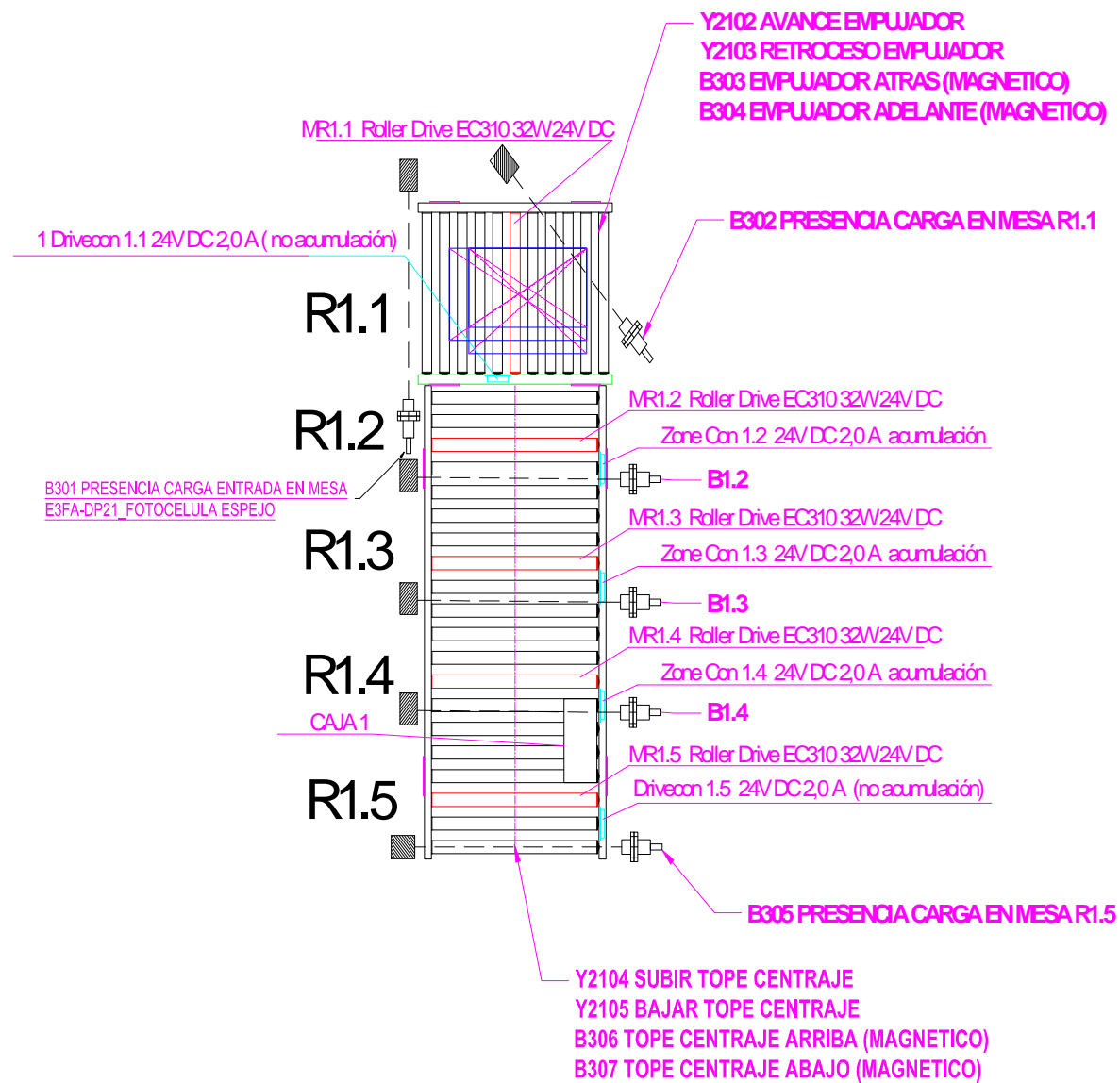
ESTE DOCUMENTO ES PROPIEDAD DE AUTOMATIZACION Y SOLUCIONES AVANZADAS.
 ESTA PROHIBIDA TODA REPRODUCCION TOTAL O PARCIAL SIN AUTORIZACION.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



 AUTOMATIZACION Y SOLUCIONES AVANZADAS S.L.	DIBUJADO :				Doc nº : 15111211	PALETIZADO ROLLOS DE PAPEL PLANTA GENERAL	FOLIO 5 ◀ 2 6 ▶
	VERIFICADO :						
	FECHA DE CREATION : Octubre 2015						
	INDICE	FECHA	COPIA TALLER	FPR			
			MODIFICACION	DES.			

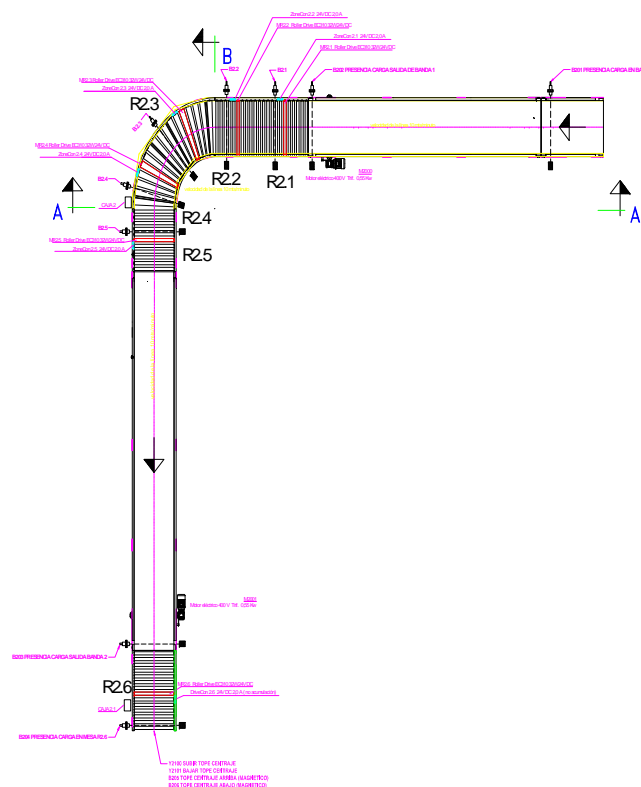
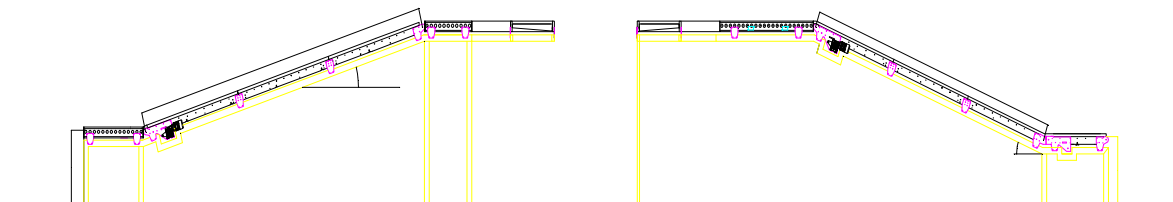
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----




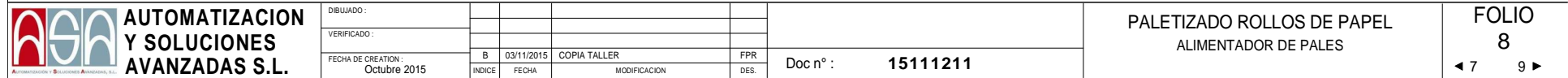
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

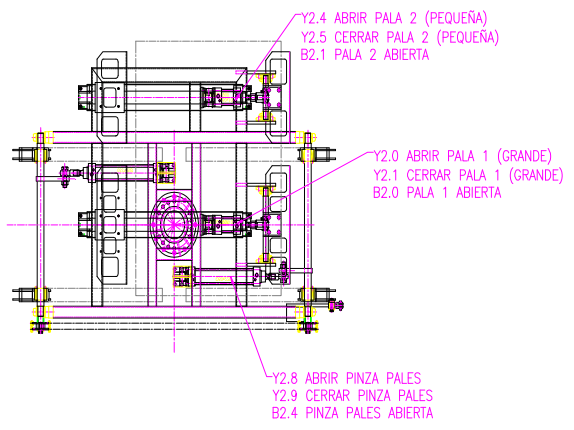
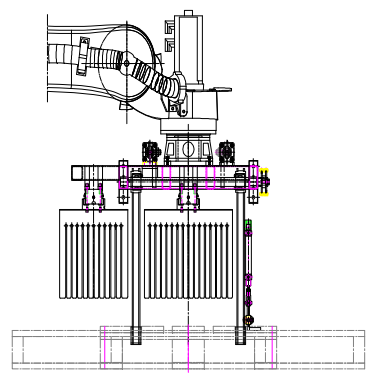
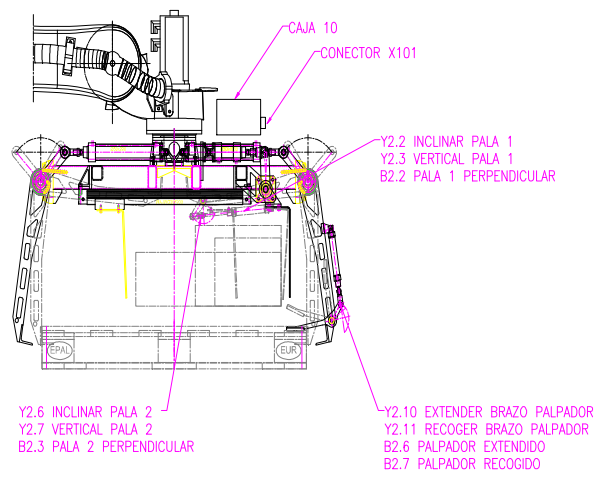
VISTA POR- B

VISTA POR- A

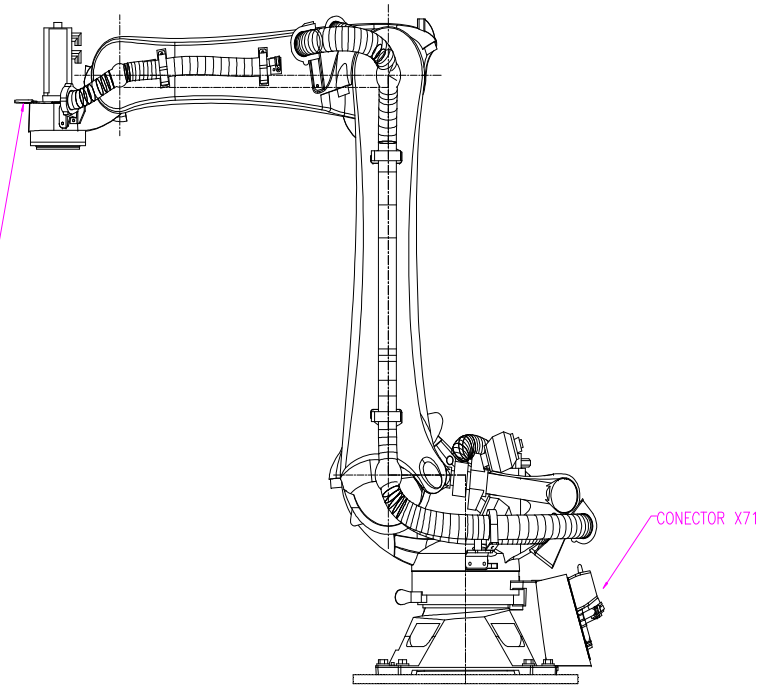


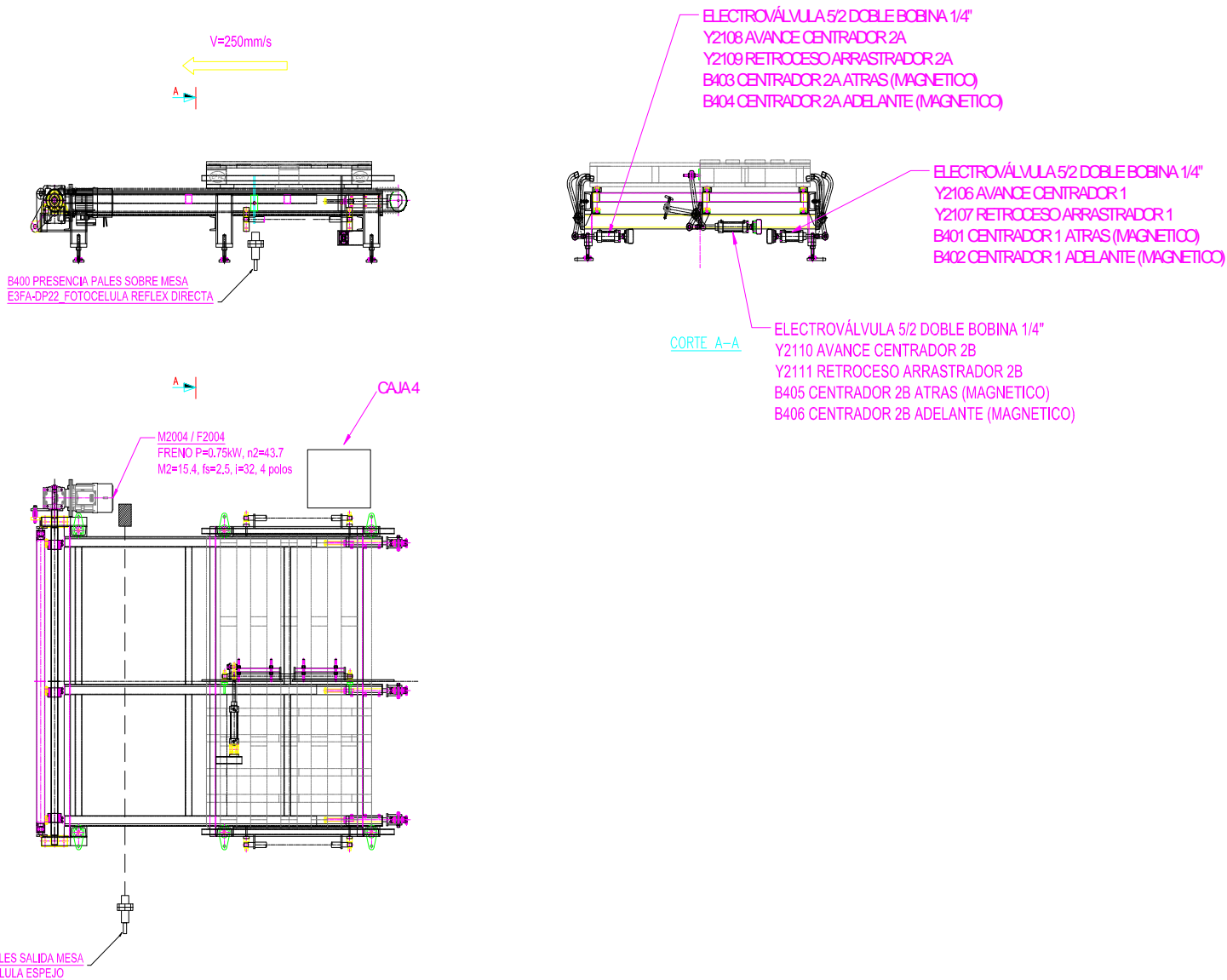
 <small>AUTOMATIZACION Y SOLUCIONES AVANZADAS, S.L.</small>	AUTOMATIZACION Y SOLUCIONES AVANZADAS S.L.				DIBUJADO :						Doc nº : 15111211	PALETIZADO ROLLOS DE PAPEL LINEA 2 DE TRANSPORTE PRODUCTO DOMESTICO	FOLIO 7 ◀ 6 8 ▶
	VERIFICADO :												
	FECHA DE CREATION : Octubre 2015				B 03/11/2015		COPIA TALLER		FPR				
	INDICE		FECHA		MODIFICACION		DES.						



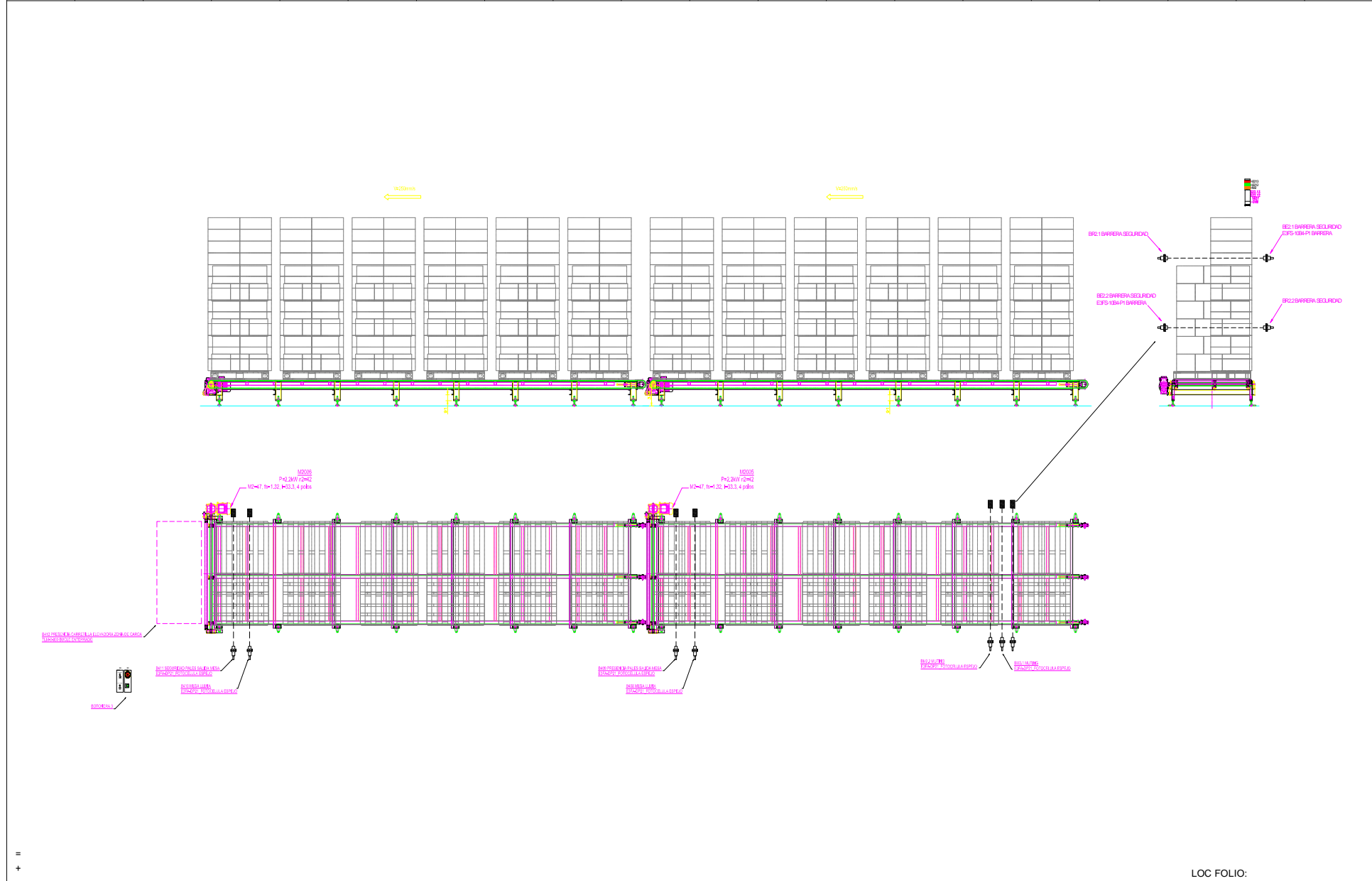


CONECTOX X91





1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----



PROGRAMA DE ROBOT

```

DEFDAT $CONFIG
BASISTECH GLOBALS
AUTOEXT GLOBALS
BackupManagerConfig
Conveyor
USER GLOBALS
;*****
;Make your modifications -ONLY- here
;*****
;=====
; Userdefined Types
;=====

;=====
; Userdefined Externals
;=====

;=====
; Userdefined Variables
;=====

;=====
; VARIABLES AUXILIARES PARA CONVERSIÓN DE SIGNO DE DATOS NUMÉRICOS
;=====

INT DATO_IN;Entrada de Dato a enviar o recibido de PLC
INT DATO_OUT;Dato convertido para enviar o recibido del PLC

;=====
; COTAS Z DE SEGURIDAD
;=====
;COTA Z SEGURA (BASE = $NULLFRAME)
INT COTA_Z_SEGURA=1890; Altura para llevar robot a posición HOME

;ALTURA SEGURA PUNTO DE ENTRADA ALIMENTADOR PALETS
INT ALT_SEG ALIM=2300;Cota Z donde el robot no arrolla la torre de palets

;ALTURA SEGURA PARA EMPEZAR A BAJAR EN LENTA A PALPAR PALET
INT ALT_SEG_BAJAR_LENTA=75;Aplicada a la última medida tomada de posición de palet

;ALTURA PARA SUBIR AL COGER UN PALET ANTES DE DEJARLO
INT ALT_SEG_SALIDA ALIM=200

;DISTANCIA EJE X DEL PUNTO DE ESPERA A PERMISO ENTRADA EN MESA PALETIZADO
INT DIST_PAL_ESPERA=1500

;ALTURA SEGURA PUNTO ENTRADA A MESA DE PALETIZADO
INT ALT_SEG_PALETIZADO=300

;ALTURA PUNTO PARA EL PRECENTRADO DEL PALET A DEPOSITA EN MESA PALETIZADO
INT ALT_CENTRADOR=100

;ALTURA MINIMA PARA SALIR DESPUÉS DE DEJAR PALET EN MESA PALETIZADO
INT ALT_SEG_SALIDA_PAL=1000

;ALTURA MINIMA DE BAJAR CON LOS OFFSET
INT ALT_MIN_DEJ_PAQUETE=325

;ALTURA DEL PUNTO PREVIO AL DEPOSITADO
INT ALT_PRE_FINAL=50

;COTAS Z MÍNIMAS PUNTO ENTRADA/SALIDA ZONA MESA PALETIZADO
REAL Z_MIN_ENT_ZONA_PALETIZ;Cota Z mínima entrada Mesa Paletizado para dejar paquete
REAL Z_MIN_SAL_ZONA_PALETIZ;Cota Z mínima salida Mesa Paletizado tras dejar paquete

;COTAS Z MÍNIMAS ALIMENTADOR PALETS
REAL Z_MIN_SAL ALIM_PAL;Cota Z mínima salida Alimentador Palets

;=====

```

```

; VARIABLES DE CONTROL INTERNO
;=====
;PERMISO MOVIMIENTO ROBOT
BOOL PER_MOV_ROBOT

;ROBOT DETENIDO POR ENTRADA EXTERNA
BOOL ROBOT_DETENIDO=FALSE

;INDICA QUE LAS GARRAS VAN A CARGAR PALET TIPO 1 (EUROPALET)
BOOL EUROPALET

;INDICA ORDEN VOLVER A CARGAR PAQUETE TRAS DEPOSITAR EN PALET
BOOL VOLVER_A_CARGAR=FALSE

;INDICA VOLVER A LÍNEA 1 POR COTA PUNTO ENTRADA TRAS DEPOSITAR EN PALET
BOOL IR_P_COJ1_ENTRAD=TRUE
;INDICA VOLVER A LÍNEA 2 POR COTA PUNTO ENTRADA TRAS DEPOSITAR EN PALET
BOOL IR_P_COJ2_ENTRAD=FALSE

;INDICA PAQUETE PROCEDENTE DE LÍNEA 1 PARA DEPOSITAR EN PALET
BOOL PAQUETE_L1=FALSE
;INDICA PAQUETE PROCEDENTE DE LÍNEA 2 PARA DEPOSITAR EN PALET
BOOL PAQUETE_L2=FALSE

;INDICADORES PRIMER CÁLCULO TRAYECTORIA CARGA-DESCARGA PAQUETES
BOOL INI_CALC_L1;Carga-Descarga Paquetes de Línea 1
BOOL INI_CALC_L2;Carga-Descarga Paquetes de Línea 2

;=====
; STRUCT "PUNTOS": VARIABLE DATOS RECIBIDOS DEL PLC PARA DEPOSITAR PAQUETE EN PALET
;=====

;ESTRUCTURA DE DATOS A RECIBIR DEL PLC
STRUC PUNTOS BOOL GIRO_LINEA1, INT NUM_CAPA, NUM_PAQUETE, X_COGER1_N, Y_COGER1_N,
X_COGER1_G, Y_COGER1_G, OFF_Z_COGIDA, OFF_X_APROX, OFF_Y_APROX, H_PAQUETE, E6POS
POS_DESTINO

;VARIABLE DONDE SE RECIBEN LOS DATOS DEL PLC
DECL PUNTOS PUNTO_DESTINO

;=====
;PUNTOS AUX. PARA LAS BASES
;=====
DECL FRAME BASE_ALIMENT;Base Alimentador Palets
DECL FRAME BASE_PALET;Base Mesa Paletizado
DECL FRAME BASE_LINEA1;Base recogida en Línea 1
DECL FRAME BASE_LINEA2;Base recogida en Línea 2

;=====
; EIP: PLC >>> ROBOT
;=====

;PERMISO MOVIMIENTO ROBOT (SÓLO EN MODO AUTOMÁTICO EXTERNO)
SIGNAL PERMISO_MOV_ROB $IN[21]

;-----
; ESTADO SENSORES PINZA PALETIZADO
;-----
SIGNAL PINZA_ABIERTA $IN[25];Pinza paquetes abierta
SIGNAL PAQUETE_COGIDO $IN[26];Pinza cargada con paquete (pinza cerrada)

SIGNAL GARRA_ABIERTA $IN[27];Garra palets abierta
SIGNAL GARRA_CERRADA $IN[28];Garra palets cerrada (recogida)
SIGNAL PALET_COGIDO $IN[29];Garras cargadas con palet

SIGNAL PALPADOR_EXTENDIDO $IN[30];Brazo palpador palets extendido
SIGNAL PALPADOR_RECOGIDO $IN[31];Brazo palpador palets recogido
SIGNAL PALET_DETECTADO $IN[32];Palet palpado (detectado)

```

```

;-----
; PERMISOS ACCESO A PUNTOS DE LA TRAYECTORIA
;-----
SIGNAL PERMISO_CARGAR_PAQUETE $IN[33];Permiso cargar paquete de Línea 1 ó 2
SIGNAL COGIDA_LINEA1_GIRADA $IN[34];Cargar paquete en Línea 1 con pinza girada 180°
SIGNAL PERMISO_CARGAR_PALET $IN[35];Permiso cargar palet
SIGNAL PERMISO_ENTRAR_PALET $IN[36];Permiso ir hasta punto predepositado palet
SIGNAL PERMISO_DEJAR_PALET $IN[37];Permiso depositar palet sobre Mesa Paletizado
SIGNAL PERMISO_DEJAR_PAQUETE $IN[38];Permiso depositar paquete sobre el palet

;-----
; ORDENES DE TAREAS QUE DEBE EJECUTAR EL ROBOT
;-----
SIGNAL RESET_POS_PALET $IN[40];Orden iniciar medición altura torre palets
SIGNAL ROBOT_HOME $IN[41];Orden ir a posición HOME
SIGNAL TIPO_EUROPALET $IN[42];Orden ir a cargar palet tipo 1 (Europalet)
SIGNAL TIPO_DOBLEPALET $IN[43];Orden ir a cargar palet tipo 2 (Paleta Grande)
SIGNAL CARGAR_LINEA1 $IN[44];Orden ir a cargar paquete a Línea 1
SIGNAL CARGAR_LINEA2 $IN[45];Orden ir a cargar paquete a Línea 2

SIGNAL ABORTAR_TAREA $IN[47];Orden abortar tarea en curso
SIGNAL PETICION_ACCESO $IN[48];Petición de acceso al recinto vallado

SIGNAL CONF_PALET_DEPOSITADO $IN[55];Enterado del PLC que se ha depositado palet en la
Mesa de Paletizado
SIGNAL CONF_PAQUETE_DEPOSITADO $IN[56];Enterado del PLC que se ha depositado paquete en
el palet

SIGNAL DATOS_OK $IN[64];Los Datos enviados al Robot son válidos

;-----
; PARÁMETROS
;-----
SIGNAL VEL_ROBOT $IN[65] TO $IN[80];Consigna Velocidad General (%)

SIGNAL CAPA_DESTINO $IN[81] TO $IN[96];No. Capa destino depositado paquete sobre el palet
SIGNAL PAQUETE_DESTINO $IN[97] TO $IN[112];No. Paquete destino depositado paquete sobre
el palet
SIGNAL ALT_PAQUETE $IN[113] TO $IN[128];Altura del paquete (mm)

SIGNAL X_PUNTO_COGER1_NORMAL $IN[129] TO $IN[144];X Cargar Paquete Línea 1 Capa Normal
SIGNAL Y_PUNTO_COGER1_NORMAL $IN[145] TO $IN[160];Y Cargar Paquete Línea 1 Capa Normal
SIGNAL X_PUNTO_COGER1_GIRADO $IN[161] TO $IN[176];X Cargar Paquete Línea 1 Capa Girada
SIGNAL Y_PUNTO_COGER1_GIRADO $IN[177] TO $IN[192];Y Cargar Paquete Línea 1 Capa Girada

SIGNAL OFFSET_Z_CARGA $IN[193] TO $IN[208];Offset Z Punto Carga Paquete de Línea 1 ó 2
(mm)

SIGNAL X_PUNTO $IN[209] TO $IN[224];Coordenada X depositado paquete (mm)
SIGNAL Y_PUNTO $IN[225] TO $IN[240];Coordenada Y depositado paquete (mm)
SIGNAL Z_PUNTO $IN[241] TO $IN[256];Cota Z depositado paquete (mm)
SIGNAL A_PUNTO $IN[257] TO $IN[272];Giro A depositado paquete (°)
SIGNAL OFF_X $IN[273] TO $IN[288];Offset X punto predepositado paquete (mm)
SIGNAL OFF_Y $IN[289] TO $IN[304];Offset Y punto predepositado paquete (mm)

;SIGNAL $IN[305] TO $IN[320]
;SIGNAL $IN[321] TO $IN[336]
;SIGNAL $IN[337] TO $IN[352]
;SIGNAL $IN[353] TO $IN[368]
;SIGNAL $IN[369] TO $IN[384]
;SIGNAL $IN[385] TO $IN[400]
;SIGNAL $IN[401] TO $IN[416]
;SIGNAL $IN[417] TO $IN[432]
;SIGNAL $IN[433] TO $IN[448]
;SIGNAL $IN[449] TO $IN[464]

;=====

```

```

; EIP: ROBOT >>> PLC
;=====

;-----
; SALIDAS ACTUADORES PINZA PALETIZADO
;-----
SIGNAL ABRIR_PINZA $OUT[25];Orden abrir pinzas paquetes
SIGNAL CERRAR_PINZA $OUT[26];Orden cerrar pinzas paquetes

SIGNAL ABRIR_GARRA $OUT[27];Orden abrir garras palets
SIGNAL CERRAR_GARRA_TOMAR_PALET $OUT[28];Orden cerrar garras para cargar palet
SIGNAL CERRAR_GARRA $OUT[29];Orden cerrar garras para recogerlas

SIGNAL EXTENDER_PALPADOR $OUT[30];Orden extender brazo palpador palets
SIGNAL RECOGER_PALPADOR $OUT[31];Orden cerrar brazo palpador palets

SIGNAL ANULAR_PALAS_CORTAS $OUT[32];Anular palas cortas pinza paquetes

;-----
; ACUSES DE RECIBO ÓRDENES SOLICITADAS AL ROBOT
;-----
SIGNAL ENTERADO_PAQUETE_CARGADO $OUT[33];Enterado pinzas cargadas con paquete
SIGNAL ENTERADO_PALET_CARGADO $OUT[35];Enterado garras cargadas con palet

SIGNAL PALET_PRE_DEPOSITADO $OUT[36];Robot situado en punto de predepositado de palet

SIGNAL RESET_HECHO $OUT[40];Enterado realizar medición altura torre palets

SIGNAL PROG_HOME $OUT[41];Enterado orden ir a posición HOME
SIGNAL PROG_EPALET $OUT[42];Enterado orden ir a cargar palet tipo 1 (Europalet)
SIGNAL PROG_DPALET $OUT[43];Enterado orden ir a cargar palet tipo 2 (Paleta Grande)
SIGNAL PROG_LINEA1 $OUT[44];Enterado orden ir a cargar paquete a Línea 1
SIGNAL PROG_LINEA2 $OUT[45];Enterado orden ir a cargar paquete a Línea 2

SIGNAL TAREA_ABORTADA $OUT[47];Enterado orden abortar tarea en curso

SIGNAL PERMISO_ACCESO $OUT[48];Permiso acceso al recinto vallado

;-----
; INTERFERENCIA CON OTROS MECANISMOS
;-----

SIGNAL ROBOT_LINEA1 $OUT[49];Robot interfiere en área de Línea 1
SIGNAL ROBOT_LINEA2 $OUT[50];Robot interfiere en área de Línea 2
SIGNAL ROBOT_ALIMENTADOR $OUT[51];Robot interfiere en área del Alimentador de Palets
SIGNAL ROBOT_MESA $OUT[52];Robot interfiere en área de la Mesa de Paletizado

SIGNAL PALET_DEPOSITADO $OUT[55];Palet depositado sobre Mesa de Paletizado
SIGNAL PAQUETE_DEPOSITADO $OUT[56];Paquete depositado sobre el palet

SIGNAL PUNTO_CARGAR_PALET $OUT[57];Robot en punto de carga de palet
SIGNAL PUNTO_CARGAR_PAQUETE $OUT[58];Robot en punto de carga de paquete
SIGNAL PUNTO_DESCAR_PAQUETE $OUT[59];Robot en punto de descarga de paquete

;-----
; VALORES ACTUALES
;-----

SIGNAL VEL_ACT_ROBOT $OUT[65] TO $OUT[80];Velocidad General actual (%)
SIGNAL ECO_CAPA_DESTINO $OUT[81] TO $OUT[96];Eco de No. Capa Destino de descarga
SIGNAL ECO_PAQUETE_DESTINO $OUT[97] TO $OUT[112];Eco de No. Paquete Destino de descarga

;=====
; PUNTOS AUXILIARES DEJAR_PAQUETE
;=====

;BASE MESA PALETIZADO

```

```

DECL E6POS P_DEJ_ENTRAD;Punto de entrada zona por encima del palet
DECL E6POS P_DEJ_VOLVER;Punto de salida por encima del palet para volver a cargar en L1
DECL E6POS P_DEJ_APROX;Punto de aproximación
DECL E6POS P_DEJ_PREVIO;Punto previo a dejar paquete (offset de separación)
DECL E6POS P_DEJ_FINAL;Punto final dejada del paquete sobre el palet
DECL E6POS P_DEJ_SALIDA;Punto salida tras depositar paquete sobre el palet

;=====
; PUNTOS AUXILIARES COGER1_PAQUETE
;=====

;BASE CARGA PAQUETE EN LINEA1
DECL E6POS P_COJ1_ENTRADA;** NO USADO (RESERVADO) **
DECL E6POS P_COJ1_ESPERA
DECL E6POS P_COJ1_FINAL
DECL E6POS P_COJ1_SALIDA

;PUNTO COGIDA MEDIDO DESDE ROBOT
DECL E6POS XP_COJ_LINEA1={X 67.0,Y 100.0,Z -153.0,A -178.0,B 0.0,C -180.0,S 2,T 11,E1
0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}
DECL E6POS XP_COJ_LINEA1_GIRO={X 592.0,Y -372.0,Z -153.0,A 0.0,B 0.0,C -180.0,S 2,T 35,E1
0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}

;=====
; PUNTOS AUXILIARES COGER2_PAQUETE
;=====

;BASE CARGA PAQUETE LINEA2
DECL E6POS P_COJ2_ENTRADA
DECL E6POS P_COJ2_ESPERA
DECL E6POS P_COJ2_FINAL
DECL E6POS P_COJ2_SALIDA

;PUNTO COGIDA MEDIDO DESDE ROBOT
DECL E6POS XP_COJ_LINEA2={X 688.0,Y -293.0,Z -75.0,A -88.0,B 0.0,C 180.0,S 2,T 10,E1
0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}

;=====
; PUNTOS AUXILIARES COGER_PALET
;=====

;BASE ALIMENTADOR DE PALETS
DECL E6POS P_PRE ALIM={X 1351.0,Y -852.0,Z 2298.0,A -116.0,B 0.0,C -180.0,S 2,T 3,E1
0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}
DECL E6POS P_POS_PALET={X 138.0,Y 1.0,Z 160.749802,A -89.0,B -8.47168418E-13,C 180.0,S
'B0010',T 'B1011',E1 0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}

;PUNTOS COGER PALETS MEDIDOS DESDE EL ROBOT
DECL E6POS XP ALIM_FINAL1={X 138.0,Y 1.0,Z -2.0,A -89.0,B 0.0,C 180.0,S 6,T 19,E1 0.0,E2
0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}
DECL E6POS XP ALIM_FINAL2={X -259.0,Y 1.0,Z -2.0,A -89.0,B 0.0,C -180.0,S 6,T 51,E1
0.0,E2 0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}

;BASE ALIMENTADOR PALETS
DECL E6POS P ALIM_ENTRAD
DECL E6POS P ALIM_SALIDA
DECL E6POS P ALIM_FINAL

;BASE MESA PALETIZADO
DECL E6POS P PAL_ESPERA
DECL E6POS P PAL_APROX
DECL E6POS P PAL_PREP
DECL E6POS P PAL_FINAL

;PUNTOS DEJAR PALETS MEDIDOS DESDE EL ROBOT
DECL E6POS XP_DEJ_EPALET={X 496.0,Y 620.0,Z 218.0,A 0.0,B 0.0,C -180.0,S 2,T 3,E1 0.0,E2
0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}
DECL E6POS XP_DEJ_DPALET={X 496.0,Y 214.0,Z 218.0,A 0.0,B 0.0,C -180.0,S 2,T 3,E1 0.0,E2
0.0,E3 0.0,E4 0.0,E5 0.0,E6 0.0}

```

DECL E6POS P_PAL_SALIDA

;PUNTO DE SEGURIDAD A LA ALTURA DEL PUNTO HOME

DECL E6POS P_SEG_HOME

ENDDAT


```

DEF SPS ( )
DECLARATIONS
INI

    LOOP
        WAIT FOR NOT($POWER_FAIL)
        TORQUE_MONITORING()
BACKUPMANAGER PLC
USER PLC
    ;Make your modifications here

;-----
;ORDEN MARCHA ROBOT (PERMISO MOVIMIENTO)
;Acciona Interrupción 3 -> IR_STOPM(); Sección INI en "principal.src"
;Sólo en Modo Automático Externo
;-----
IF $MODE_OP == #EX THEN
    PER_MOV_ROBOT = PERMISO_MOV_ROB
ELSE
    PER_MOV_ROBOT = TRUE
ENDIF

;-----
;VELOCIDAD GENERAL DEL ROBOT
;-----
$OV_PRO = VEL_ROBOT ;Del PLC: Consigna velocidad (%)
VEL_ACT_ROBOT = $OV_PRO ;Al PLC: Velocidad actual del Robot (%)

;-----
;ORDEN TOMAR NUEVA MEDIDA DE ALTURA EN LA TORRE DE PALETS PARA COGER
;-----
IF RESET_POS_PALET THEN
    P_POS_PALET = {X 0,Y 0,Z 0,A 0,B 0,C 0}
    RESET_HECHO = TRUE ;Acuse de la orden recibida
ELSE
    RESET_HECHO = FALSE ;Fin acuse cuando cesa la orden
ENDIF

;=====
;TRASPASO DE LOS DATOS RECIBIDOS DEL PLC A LA ESTRUCTURA PUNTO_DESTINO
;=====
PUNTO_DESTINO.NUM_CAPA = CAPA_DESTINO ;No. Capa destino depositado paquete
PUNTO_DESTINO.NUM_PAQUETE = PAQUETE_DESTINO ;No. paquete destino depositado
PUNTO_DESTINO.GIRO_LINEAL = COGIDA_LINEAL_GIRADA ;Coger paquete girado Línea 1
PUNTO_DESTINO.H_PAQUETE = ALT_PAQUETE; Altura del paquete

;-----
; CONVERSION DE SIGNO DE DATOS RECIBIDOS DEL PLC
;-----
DATO_IN = X_PUNTO
CONVERTIR_E()
PUNTO_DESTINO.POS_DESTINO.X = DATO_OUT ;"X" Punto destino depositado paquete

DATO_IN = Y_PUNTO
CONVERTIR_E()
PUNTO_DESTINO.POS_DESTINO.Y = DATO_OUT ;"Y" Punto destino depositado paquete

DATO_IN = Z_PUNTO
CONVERTIR_E()
PUNTO_DESTINO.POS_DESTINO.Z = DATO_OUT ;"Z" Punto destino depositado paquete

DATO_IN = A_PUNTO
CONVERTIR_E()
PUNTO_DESTINO.POS_DESTINO.A = DATO_OUT ;"A" Punto destino depositado paquete

DATO_IN = OFF_X
CONVERTIR_E()
PUNTO_DESTINO.OFF_X_APROX = DATO_OUT ;Offset "X" Punto aproximación depositado

```

```

DATO_IN = OFF_Y
CONVERTIR_E()
PUNTO_DESTINO.OFF_Y_APROX = DATO_OUT ;Offset "Y" Punto aproximación depositado

DATO_IN = X_PUNTO_COGER1_NORMAL
CONVERTIR_E()
PUNTO_DESTINO.X_COGER1_N = DATO_OUT ;"X" Punto recogida paquete L1 (normal)

DATO_IN = Y_PUNTO_COGER1_NORMAL
CONVERTIR_E()
PUNTO_DESTINO.Y_COGER1_N = DATO_OUT ;"Y" Punto recogida paquete L1 (normal)

DATO_IN = X_PUNTO_COGER1_GIRADO
CONVERTIR_E()
PUNTO_DESTINO.X_COGER1_G = DATO_OUT ;"X" Punto recogida paquete L1 (pinza girada 180°)

DATO_IN = Y_PUNTO_COGER1_GIRADO
CONVERTIR_E()
PUNTO_DESTINO.Y_COGER1_G = DATO_OUT ;"Y" Punto recogida paquete L1 (pinza girada 180°)

DATO_IN = OFFSET_Z_CARGA
CONVERTIR_E()
PUNTO_DESTINO.OFF_Z_COGIDA = DATO_OUT ;Offset "Z" Punto recogida paquete L1

;=====
;ENVÍO DE DATOS AL PLC
;=====
ECO_CAPA_DESTINO = PUNTO_DESTINO.NUM_CAPA ;No. capa destino depositado paquete
ECO_PAQUETE_DESTINO = PUNTO_DESTINO.NUM_PAQUETE ;No. paquete destino depositado

;=====
;RESET ACUSES DE ORDEN RECIBIDA DEL PLC
;=====
;SI NO HAY ORDEN DE CARGAR EN LÍNEA 1
IF NOT CARGAR_LINEA1 THEN
    PROG_LINEA1 = FALSE
ENDIF
;SI NO HAY ORDEN DE CARGAR EN LÍNEA 2
IF NOT CARGAR_LINEA2 THEN
    PROG_LINEA2 = FALSE
ENDIF
;SI NO HAY ORDEN DE CARGAR EUROPALET
IF NOT TIPO_EUROPALET THEN
    PROG_EPALET = FALSE
ENDIF
;SI NO HAY ORDEN DE CARGAR PALETA GRANDE
IF NOT TIPO_DOBLEPALET THEN
    PROG_DPALET = FALSE
ENDIF
;SI NO HAY ORDEN DE IR A POSICIÓN "HOME"
IF NOT ROBOT_HOME THEN
    PROG_HOME = FALSE
ENDIF
;SI NO HAY ORDEN DE ABORTAR TAREA EN CURSO
IF NOT ABORTAR_TAREA THEN
    TAREA_ABORTADA = FALSE
ENDIF

ENDLOOP

```

```
DEF  abortar_tarea ( )

;DESACTIVAR LA INTERRUPCIÓN (UNA VEZ ATENDIDA)
INTERRUPT OFF 2

;INFORME ATENDIDA ORDEN ABORTAR TAREA EN CURSO
TAREA_ABORTADA = TRUE

;DETENER EL MOVIMIENTO DEL ROBOT
BAS(#ACC_PTP, 50)
BAS(#ACC_CP, 50)
BRAKE

$BASE = $NULLFRAME
$TOOL = TOOL_DATA[1]

;PRIMER MOVIMIENTO -> POSICIÓN ACTUAL
PTP $POS_ACT

;CÁLCULO PUNTO ASCENSO A COTA Z SEGURA
P_SEG_HOME = $POS_ACT
P_SEG_HOME.Z = COTA_Z_SEGURA

;SUBIR LINEAL A COTA SEGURA
BAS(#VEL_CP, 1)
LIN P_SEG_HOME

;RESET INICIAL
inicio()

;HABILITAR DE NUEVO LA INTERRUPCIÓN
INTERRUPT ON 2
RESUME
END
```

```

DEF buscar_palet ( )
CONTINUE
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVAR INTERRUPCIÓN PALPADOR DE PALET
INTERRUPT ON 4

;-----
;SI SE HA COGIDO UN PALET PREVIO -> BAJAR RÁPIDO A POSICIÓN CONOCIDA
;-----
IF P_POS_PALET.Z <> 0 THEN
    P_POS_PALET.Z = P_POS_PALET.Z + ALT_SEG_BAJAR_LENTA
    BAS(#VEL_CP, 2)
    BAS(#ACC_CP, 100)
    $APO.CDIS = 20
    LIN P_POS_PALET C_DIS
ENDIF

;-----
;TERMINAR DESCENSO EN VELOCIDAD LENTA ESPERANDO LA INTERRUPCIÓN DEL PALPADOR
;-----
BAS(#VEL_CP, 0.2)
BAS(#ACC_CP, 100)
LIN P_ALIM_FINAL

WAIT SEC 0
END

```

```

DEF calc_ptos_lineal ( )

;VARIABLES LOCALES
DECL INT dZ_COJ1_ESPERA
DECL INT dZ_COJ1_SALIDA
DECL INT dZ_DEJ_APROX

;ASIGNACIÓN VALOR VARIABLES LOCALES
;Suplemento Cota Z punto de espera hasta permiso para coger paquete
dZ_COJ1_ESPERA = 50
;Suplemento Cota Z sobre el punto de salida tras cargar paquete
dZ_COJ1_SALIDA = 50
;Suplemento Cota Z sobre el punto de aproximación para dejar paquete
dZ_DEJ_APROX = 75

;*****
;CÁLCULO DE LOS PUNTO DE TRAYECTORIA DE CARGA-DESCARGA LÍNEA 1
;*****

;=====
;CÁLCULO PUNTOS TRAYECTORIA SALIDA DEPOSITADO PAQUETE ACTUAL
;(Antes de calcular punto del próximo depositado)
;=====

IF NOT INI_CALC_L1 THEN
    ;COORDENADAS PUNTO DE SALIDA TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_SALIDA = P_DEJ_FINAL
    P_DEJ_SALIDA.Z = P_DEJ_APROX.Z

    ;COORDENADAS PUNTO VOLVER A CARGAR L1 TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_VOLVER = P_DEJ_ENTRAD
    P_DEJ_VOLVER.Z = P_DEJ_SALIDA.Z

    ;COTA MÍNIMA PARA VOLVER POR PUNTO ENTRADA MESA PALETIZADO TRAS DEPOSITAR
    Z_MIN_SAL_ZONA_PALETIZ = P_COJ1_ENTRADA.Z + BASE_DATA[3].Z - BASE_DATA[2].Z

    ;COTA Z REGRESO AL PUNTO DE ENTRADA MESA PALETIZADO
    IR_P_COJ1_ENTRAD = FALSE
    IF P_DEJ_VOLVER.Z < Z_MIN_SAL_ZONA_PALETIZ THEN
        P_DEJ_VOLVER.Z = Z_MIN_SAL_ZONA_PALETIZ
        ;Si cota salida depositado paquete menor que cota entrada cargar paquete L1
        ;volver a cargar por cota de entrada
        IR_P_COJ1_ENTRAD = TRUE
    ENDIF
ENDIF

;=====
;CÁLCULO PUNTOS TRAYECTORIA ZONA RECOGIDA PRÓXIMO PAQUETE LÍNEA 1
;=====
;-----
;COORDENADAS DEL PUNTO FINAL DE CARGA EN LÍNEA 1
;-----
CONTINUE
IF PUNTO_DESTINO.GIRO_LINEA1 THEN
    P_COJ1_FINAL = XP_COJ_LINEA1_GIRO
    P_COJ1_FINAL.X = PUNTO_DESTINO.X_COGER1_G
    P_COJ1_FINAL.Y = PUNTO_DESTINO.Y_COGER1_G
    P_COJ1_FINAL.Z = XP_COJ_LINEA1_GIRO.Z + PUNTO_DESTINO.OFF_Z_COGIDA
CONTINUE
ELSE
    P_COJ1_FINAL = XP_COJ_LINEA1
    P_COJ1_FINAL.X = PUNTO_DESTINO.X_COGER1_N
    P_COJ1_FINAL.Y = PUNTO_DESTINO.Y_COGER1_N
    P_COJ1_FINAL.Z = XP_COJ_LINEA1.Z + PUNTO_DESTINO.OFF_Z_COGIDA
ENDIF

;-----
;COORDENADAS DEL PUNTO DE ESPERA HASTA PERMISO PARA CARGAR EN LÍNEA 1
;-----

```

```

P_COJ1_ESPERA = P_COJ1_FINAL
CONTINUE
P_COJ1_ESPERA.Z = P_COJ1_FINAL.Z + ALT_PAQUETE + dZ_COJ1_ESPERA

;-----
;COORDENADAS DEL PUNTO DE ENTRADA PARA CARGAR EN LÍNEA 1
;-----
P_COJ1_ENTRADA = P_COJ1_ESPERA
P_COJ1_ENTRADA.X = P_COJ1_ESPERA.X + 200
P_COJ1_ENTRADA.Z = P_COJ1_ESPERA.Z + 50

;-----
;COORDENADAS DEL PUNTO DE SALIDA TRAS CARGAR EN LÍNEA 1
;-----
P_COJ1_SALIDA = P_COJ1_ESPERA
P_COJ1_SALIDA.Z = P_COJ1_SALIDA.Z + dZ_COJ1_SALIDA ;Para librar Mesa de Carga

;=====
;CÁLCULO PUNTOS TRAYECTORIA DEPOSITADO EN MESA PALETIZADO
;=====
;-----
;COORDENADAS PUNTO FINAL DEPOSITADO PRÓXIMO PAQUETE SOBRE PALET
;-----
P_DEJ_FINAL = PUNTO_DESTINO.POS_DESTINO

;-----
;COORDENADAS PUNTO PREVIO DEJADA PAQUETE (CON OFFSETS DE SEPARACIÓN)
;-----
P_DEJ_PREVIO = P_DEJ_FINAL
P_DEJ_PREVIO.X = P_DEJ_FINAL.X + PUNTO_DESTINO.OFF_X_APROX
P_DEJ_PREVIO.Y = P_DEJ_FINAL.Y + PUNTO_DESTINO.OFF_Y_APROX
P_DEJ_PREVIO.Z = P_DEJ_FINAL.Z + ALT_PRE_FINAL

;-----
;COORDENADAS PUNTO DE APROXIMACIÓN (POR ENCIMA DE CAPA ACTUAL DEL PALET)
;-----
P_DEJ_APROX = P_DEJ_PREVIO
CONTINUE
P_DEJ_APROX.Z = P_DEJ_FINAL.Z + PUNTO_DESTINO.H_PAQUETE + dZ_DEJ_APROX

;-----
;COORDENADAS PUNTO ENTRADA ANTES DE LLEGAR AL AREA DEL PALET
;-----
P_DEJ_ENTRAD = P_DEJ_APROX
P_DEJ_ENTRAD.X = 1906
P_DEJ_ENTRAD.Y = -380

;COTA MÍNIMA PARA IR A PUNTO ENTRADA MESA PALETIZADO
;(Referido a la Base de Puntos de Mesa Paletizado)
CONTINUE
Z_MIN_ENT_ZONA_PALETIZ = P_COJ1_SALIDA.Z + BASE_DATA[3].Z - BASE_DATA[2].Z

;SI LA COTA Z DEL PUNTO DE APROXIMACIÓN PARA DEPOSITAR EN PALET ES
;MENOR A LA COTA ACTUAL DEL ROBOT, IR A PUNTO DE ENTRADA A ZONA PALET
;EN LA MISMA COTA QUE EL PUNTO DE APROXIMACIÓN DE DEPOSITADO

IF (P_DEJ_APROX.Z < Z_MIN_ENT_ZONA_PALETIZ) THEN
    P_DEJ_ENTRAD.Z = Z_MIN_ENT_ZONA_PALETIZ
ENDIF

;-----
;GIRO PUNTO VOLVER PARA CARGAR PRÓXIMO PAQUETE EN LÍNEA 1
;-----
P_DEJ_VOLVER.A = P_COJ1_FINAL.A + BASE_DATA[3].A - BASE_DATA[2].A

;=====
;CÁLCULO PUNTOS TRAYECTORIA DE SALIDA DEPOSITADO PAQUETE

```

```

;(Sólo cuando se inician los cálculos por primera vez)
;=====

IF INI_CALC_L1 THEN
    ;COORDENADAS PUNTO DE SALIDA TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_SALIDA = P_DEJ_FINAL
    P_DEJ_SALIDA.Z = P_DEJ_APROX.Z

    ;COORDENADAS PUNTO VOLVER A CARGAR L1 TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_VOLVER = P_DEJ_ENTRAD
    P_DEJ_VOLVER.Z = P_DEJ_SALIDA.Z

    ;COTA MÍNIMA PARA VOLVER POR PUNTO ENTRADA MESA PALETIZADO TRAS DEPOSITAR
    Z_MIN_SAL_ZONA_PALETIZ = P_COJ1_ENTRADA.Z + BASE_DATA[3].Z - BASE_DATA[2].Z

    ;COTA Z REGRESO AL PUNTO DE ENTRADA MESA PALETIZADO
    IR_P_COJ1_ENTRAD = FALSE
    IF P_DEJ_VOLVER.Z < Z_MIN_SAL_ZONA_PALETIZ THEN
        P_DEJ_VOLVER.Z = Z_MIN_SAL_ZONA_PALETIZ
        ;Si cota salida depositado paquete menor que cota entrada cargar paquete L1
        ;volver a cargar por cota de entrada
        IR_P_COJ1_ENTRAD = TRUE
    ENDIF
ENDIF

;RESET CÁLCULO INICIAL DE PUNTOS DE TRAYECTORIA CARGA-DESCARGA LÍNEA 1
INI_CALC_L1 = FALSE

END

```

```

DEF calc_ptos_linea2 ( )

;VARIABLES LOCALES
DECL INT dZ_DEJ_APROX
DECL INT dZ_COJ2_ESPERA
DECL INT dZ_COJ2_SALIDA

;ASIGNACIÓN VALOR VARIABLES LOCALES
;Suplemento Cota Z para entrar por encima del paquete para depositar
dZ_DEJ_APROX = 40
;Suplemento Cota Z sobre el punto de espera para cargar paquete en L1
dZ_COJ2_ESPERA = 50
;Suplemento Cota Z sobre el punto de salida tras cargar paquete
dZ_COJ2_SALIDA = 50

;*****
;CÁLCULO DE LOS PUNTO DE TRAYECTORIA DE CARGA-DESCARGA LÍNEA 2
;*****

;=====
;CÁLCULO PUNTOS TRAYECTORIA SALIDA DEPOSITADO PAQUETE ACTUAL
;(Antes de calcular punto del próximo depositado)
;=====

IF NOT INI_CALC_L2 THEN
    ;COORDENADAS PUNTO DE SALIDA TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_SALIDA = P_DEJ_FINAL
    ;P_DEJ_SALIDA.Z = P_DEJ_APROX.Z
    P_DEJ_SALIDA.Z = P_DEJ_FINAL.Z + PUNTO_DESTINO.H_PAQUETE + dZ_DEJ_APROX

    ;COORDENADAS PUNTO VOLVER A CARGAR L2 TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_VOLVER = P_DEJ_ENTRAD
    P_DEJ_VOLVER.Z = P_DEJ_SALIDA.Z

    ;COTA MÍNIMA PARA VOLVER POR PUNTO ENTRADA MESA PALETIZADO TRAS DEPOSITAR
    Z_MIN_SAL_ZONA_PALETIZ = P_COJ2_ENTRADA.Z + BASE_DATA[4].Z - BASE_DATA[2].Z

    ;COTA Z REGRESO AL PUNTO DE ENTRADA MESA PALETIZADO
    IR_P_COJ2_ENTRAD = FALSE
    IF P_DEJ_VOLVER.Z < Z_MIN_SAL_ZONA_PALETIZ THEN
        P_DEJ_VOLVER.Z = Z_MIN_SAL_ZONA_PALETIZ
        ;Si cota salida depositado paquete menor que cota entrada cargar paquete L2
        ;volver a cargar por cota de entrada
        IR_P_COJ2_ENTRAD = TRUE
    ENDIF
ENDIF

;=====
;CÁLCULO PUNTOS TRAYECTORIA ZONA RECOGIDA PAQUETE LÍNEA 2
;=====
;-----
;COORDENADAS DEL PUNTO FINAL DE CARGA EN LÍNEA 2
;-----
P_COJ2_FINAL = XP_COJ_LINEA2
P_COJ2_FINAL.Z = XP_COJ_LINEA2.Z + PUNTO_DESTINO.OFF_Z_COGIDA

;-----
;COORDENADAS DEL PUNTO DE ESPERA HASTA PERMISO PARA CARGAR EN LÍNEA 2
;-----
P_COJ2_ESPERA = P_COJ2_FINAL
CONTINUE
P_COJ2_ESPERA.Z = P_COJ2_FINAL.Z + ALT_PAQUETE + dZ_COJ2_ESPERA

;-----
;COORDENADAS DEL PUNTO DE ENTRADA PARA CARGAR EN LÍNEA 2
;-----
P_COJ2_ENTRADA = P_COJ2_ESPERA
P_COJ2_ENTRADA.Y = P_COJ2_ESPERA.Y - 200

```



```

P_COJ2_ENTRADA.Z = P_COJ2_ESPERA.Z + 50

;-----
;COORDENADAS DEL PUNTO DE SALIDA TRAS CARGAR EN LÍNEA 2
;-----
P_COJ2_SALIDA = P_COJ2_ESPERA
P_COJ2_SALIDA.Z = P_COJ2_ESPERA.Z + dZ_COJ2_SALIDA

;=====
;CÁLCULO PUNTOS TRAYECTORIA DEPOSITADO EN MESA PALETIZADO
;=====
;-----
;COORDENADAS PUNTO FINAL DEJADA PAQUETE SOBRE PALET
;-----
P_DEJ_FINAL = PUNTO_DESTINO.POS_DESTINO

;-----
;COORDENADAS PUNTO PREVIO DEJADA PAQUETE (CON OFFSETS DE SEPARACIÓN)
;-----
P_DEJ_PREVIO = P_DEJ_FINAL
P_DEJ_PREVIO.X = P_DEJ_FINAL.X + PUNTO_DESTINO.OFF_X_APROX
P_DEJ_PREVIO.Y = P_DEJ_FINAL.Y + PUNTO_DESTINO.OFF_Y_APROX
P_DEJ_PREVIO.Z = P_DEJ_FINAL.Z + ALT_PRE_FINAL

;-----
;COORDENADAS PUNTO DE APROXIMACIÓN (POR ENCIMA DE CAPA ACTUAL DEL PALET)
;-----
P_DEJ_APROX = P_DEJ_PREVIO
P_DEJ_APROX.Z = P_DEJ_FINAL.Z + PUNTO_DESTINO.H_PAQUETE + dZ_DEJ_APROX

; ** Prueba de aproximación a depositado paquete en cota baja *****
;-----
CONTINUE
IF (PUNTO_DESTINO.NUM_CAPA == 13) THEN
    P_DEJ_APROX.Z = P_DEJ_FINAL.Z + 80
ENDIF
;*****

;-----
;COORDENADAS PUNTO ENTRADA ANTES DE LLEGAR AL AREA DEL PALET
;-----
P_DEJ_ENTRAD = P_DEJ_APROX
P_DEJ_ENTRAD.X = 1906
P_DEJ_ENTRAD.Y = -380

;COTA MÍNIMA PARA IR A PUNTO ENTRADA MESA PALETIZADO
;(Referido a la Base de Puntos de Mesa Paletizado)
CONTINUE
Z_MIN_ENT_ZONA_PALETIZ = P_COJ2_SALIDA.Z + BASE_DATA[4].Z - BASE_DATA[2].Z

;SI LA COTA Z DEL PUNTO DE APROXIMACIÓN PARA DEPOSITAR EN PALET ES
;MENOR A LA COTA ACTUAL DEL ROBOT, IR A PUNTO DE ENTRADA A ZONA PALET
;EN LA MISMA COTA QUE EL PUNTO DE APROXIMACIÓN DE DEPOSITADO

IF (P_DEJ_APROX.Z < Z_MIN_ENT_ZONA_PALETIZ) THEN
    P_DEJ_ENTRAD.Z = Z_MIN_ENT_ZONA_PALETIZ
ENDIF

;=====
;CÁLCULO PUNTOS TRAYECTORIA DE SALIDA DEPOSITADO PAQUETE
;(Sólo cuando se inician los cálculos por primera vez)
;=====

IF INI_CALC_L2 THEN
    ;COORDENADAS PUNTO DE SALIDA TRAS DEPOSITADO DE PAQUETE ACTUAL
    P_DEJ_SALIDA = P_DEJ_FINAL
    ;P_DEJ_SALIDA.Z = P_DEJ_APROX.Z

```

```

P_DEJ_SALIDA.Z = P_DEJ_FINAL.Z + PUNTO_DESTINO.H_PAQUETE + dz_DEJ_APROX

;COORDENADAS PUNTO VOLVER A CARGAR L2 TRAS DEPOSITADO DE PAQUETE ACTUAL
P_DEJ_VOLVER = P_DEJ_ENTRAD
P_DEJ_VOLVER.Z = P_DEJ_SALIDA.Z

;COTA MÍNIMA PARA VOLVER POR PUNTO ENTRADA MESA PALETIZADO TRAS DEPOSITAR
Z_MIN_SAL_ZONA_PALETIZ = P_COJ2_ENTRADA.Z + BASE_DATA[4].Z - BASE_DATA[2].Z

;COTA Z REGRESO AL PUNTO DE ENTRADA MESA PALETIZADO
IR_P_COJ2_ENTRAD = FALSE
IF P_DEJ_VOLVER.Z < Z_MIN_SAL_ZONA_PALETIZ THEN
    P_DEJ_VOLVER.Z = Z_MIN_SAL_ZONA_PALETIZ
    ;Si cota salida depositado paquete menor que cota entrada cargar paquete L1
    ;volver a cargar por cota de entrada
    IR_P_COJ2_ENTRAD = TRUE
ENDIF
ENDIF

;RESET CÁLCULO INICIAL DE PUNTOS DE TRAYECTORIA CARGA-DESCARGA LÍNEA 2
INI_CALC_L2 = FALSE

END

```

```

DEF cogrer_palet ( )
CONTINUE
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPTIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPTIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVACIÓN DE INTERRUPTIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;ACUSE AL PLC DE EJECUCIÓN DE ORDEN CARGAR PALET TIPO 1 ó 2
IF EUROPALET THEN
    CONTINUE
    PROG_EPALET = TRUE ;Ejecución Tarea Cargar Palet Tipo 1 (Europalet)
    CONTINUE
    PROG_DPALET = FALSE
ELSE
    CONTINUE
    PROG_DPALET = TRUE ;Ejecución Tarea Cargar Palet Tipo 2 (Paleta Grande)
    CONTINUE
    PROG_EPALET = FALSE
ENDIF

;-----
;CÁLCULO PREVIO PUNTOS DEPOSITADO PALET EN MESA PALETIZADO
;-----
;PUNTO FINAL DEPOSITADO PALET SOBREMESA (SEGÚN TIPO DE PALET)
IF EUROPALET THEN
    P_PAL_FINAL = XP_DEJ_EPALET ;Si Garras cargadas con Europalet
ELSE
    P_PAL_FINAL = XP_DEJ_DPALET ;Si Garras crgadas con Paleta Grande
ENDIF

;PUNTO DE PREDEPOSITADO (PRECENTRADO PALET ANTES DE DEPOSITAR)
P_PAL_PREP = P_PAL_FINAL
P_PAL_PREP.Z = P_PAL_FINAL.Z + ALT_CENTRADOR

;PUNTO DE APROXIMACIÓN AL DEPOSITADO DEL PALET
P_PAL_APROX = P_PAL_FINAL
P_PAL_APROX.Z = P_PAL_FINAL.Z + ALT_SEG_PALETIZADO

;=====
;SI LA GARRA ESTÁ CARGADA CON PALET -> IR A DEPOSITAR PALET SOBRE MESA PALETIZADO
;=====
CONTINUE

```

```

IF PALET_COGIDO THEN

;CAMBIO A BASE PUNTOS MESA PALETIZADO
CONTINUE
$BASE = BASE_DATA[2]

;PUNTO DE ESPERA ANTES DE ENTRAR EN ÁREA DE MESA DE PALETIZADO
P_PAL_ESPERA = P_PAL_APROX
P_PAL_ESPERA.X = P_PAL_APROX.X - DIST_PAL_ESPERA
P_PAL_ESPERA.Z = P_ALIM_SALIDA.Z + BASE_DATA[1].Z - BASE_DATA[2].Z

;PUNTO DE SALIDA TRAS DEPOSITAR PALET EN MESA PALETIZADO
P_PAL_SALIDA = P_PAL_FINAL
P_PAL_SALIDA.Z = P_PAL_FINAL.Z + ALT_SEG_SALIDA_PAL

;*****
;TRAYECTORIA PUNTOS DEPOSITADO PALET EN MESA PALETIZADO
;*****
;-----
;IR A PUNTO DE ESPERA ENTRADA A MESA PALETIZADO (PALET EN GARRAS)
;-----
BAS(#VEL_PTP, 60)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
PTP P_PAL_ESPERA C_PTP C_DIS

;ESPERA PERMISO PARA ENTRAR CON PALET EN MESA PALETIZADO
CONTINUE
WHILE NOT PERMISO_ENTRAR_PALET
    WAIT SEC 0.01
ENDWHILE

;-----
;IR A PUNTO DE APROXIMACIÓN SOBRE MESA PALETIZADO
;-----
BAS(#VEL_PTP, 60)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
TRIGGER WHEN DISTANCE=0 DELAY=50 DO ROBOT_MESA = TRUE
PTP P_PAL_APROX C_PTP C_DIS

;-----
;IR A PUNTO PREDEPOSITADO PALET
;-----
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 50
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO CERRAR_GARRA_TOMAR_PALET = FALSE
TRIGGER WHEN DISTANCE=1 DELAY=0 DO PALET_PRE_DEPOSITADO = TRUE
LIN P_PAL_PREP C_DIS

;ESPERA PERMISO PARA DEPOSITAR PALET SOBRE MESA PALETIZADO
CONTINUE
WHILE NOT PERMISO_DEJAR_PALET
    WAIT SEC 0.01
ENDWHILE

;-----
;IR A PUNTO FINAL DEPOSITADO PALET SOBRE MESA PALETIZADO
;-----
BAS(#VEL_CP, 0.1)
BAS(#ACC_CP, 100)
LIN P_PAL_FINAL

;APERTURA DE LA GARRA PARA DEPOSITAR EL PALET
CERRAR_GARRA = FALSE
ABRIR_GARRA = TRUE

```

```

;ESPERA CONFIRMACIÓN GARRA ABIERTA (PALET DEPOSITADO)
WHILE NOT GARRA_ABIERTA
    WAIT SEC 0.01
ENDWHILE

;ENVÍA INFORME PALET DEPOSITADO SOBRE MESA PALETIZADO
PALET_DEPOSITADO = TRUE
ENTERADO_PALET_CARGADO = FALSE

;ESPERA CONFIRMACIÓN PALET DEPOSITADO SOBRE MESA PALETIZADO
WHILE NOT CONF_PALET_DEPOSITADO
    WAIT SEC 0.01
ENDWHILE
PALET_DEPOSITADO = FALSE

;-----
;IR A PUNTO DE SALIDA TRAS DEPOSITAR PALET EN MESA PALETIZADO
;-----
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 50
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO ABRIR_GARRA = FALSE
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO CERRAR_GARRA = TRUE
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO ROBOT_MESA = FALSE
LIN P_PAL_SALIDA C_DIS

;ESPERA GARRA PALET CERRADA ANTES DE CONTINUAR
CONTINUE
WHILE NOT GARRA_CERRADA
    WAIT SEC 0.01
ENDWHILE

;=====
;SI LA GARRA NO ESTÁ CARGADA CON PALET -> IR A CARGAR PALET AL ALIMENTADOR
;=====
ELSE

    ;DECLARAMOS LA INTERRUPCIÓN PARA DETECTAR PALET
    $CYCFLAG[4] = PALET_DETECTADO
    INTERRUPT DECL 4 WHEN $CYCFLAG[4] DO palet_detectado()

    ;BASE PUNTOS EN ZONA ALIMENTADOR DE PALETS
    CONTINUE
    $TOOL = TOOL_DATA[1]
    CONTINUE
    $BASE = BASE_DATA[1]

    ;-----
    ;CÁLCULO DE PUNTOS CARGA PALET EN ALIMENTADOR
    ;-----
    ;PUNTO FINAL MÁS BAJO EN LA VERTICAL DEL ALIMENTADOR (SEGÚN TIPO PALET)
    IF EUROPALET THEN
        P_ALIM_FINAL = XP_ALIM_FINAL1 ;Si se va a cargar Europalet
    ELSE
        P_ALIM_FINAL = XP_ALIM_FINAL2 ;Si se va a cargar Paleta Grande
    ENDIF

    ;PUNTO ENTRADA EN LA VERTICAL DEL ALIMENTADOR
    P_ALIM_ENTRAD = P_ALIM_FINAL
    P_ALIM_ENTRAD.Z = P_ALIM_FINAL.Z + ALT_SEG_ALIM

    ;PUNTO PREVIO A ENTRADA EN ZONA ALIMENTADOR PALETS
    P_PRE_ALIM.Z = P_ALIM_ENTRAD.Z

    ;*****
    ;TRAYECTORIA PUNTOS DEPOSITADO PALET EN MESA PALETIZADO
    ;*****

```

```

;-----
;IR AL PUNTO PREVIO AL ALIMENTADOR DE PALETS
;(En este punto preparamos garras para coger palet)
;-----
CONTINUE
BAS(#VEL_PTP, 80)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 100
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO posic_garra_cargar() prio=-1
PTP P_PRE ALIM C_PTP C_DIS

;-----
;IR AL PUNTO DE ENTRADA SOBRE LA VERTICAL DE LA TORRE DE PALETS
;-----
TRIGGER WHEN DISTANCE=1 DELAY=0 DO ROBOT_ALIMENTADOR = TRUE
BAS(#VEL_PTP, 80)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 100
PTP P ALIM ENTRAD C_PTP C_DIS

;ESPERAR CONFIRMACIÓN GARRA PREPARADA PARA BUSCAR PALET
CONTINUE
WHILE NOT (GARRA_ABIERTA AND PALPADOR_EXTENDIDO)
    WAIT SEC 0.01
ENDWHILE

;ESPERAR PERMISO PARA REALIZAR TAREA DE CARGAR PALET
CONTINUE
WHILE NOT PERMISO_CARGAR_PALET
    WAIT SEC 0.01
ENDWHILE

;=====
;RUTINA BUSCAR PALET PARA CARGAR (PALPADOR)
buscar_palet()
;=====

;CERRAR GARRA PARA CARGAR PALET LOCALIZADO
ABRIR_GARRA = FALSE
CERRAR_GARRA_TOMAR_PALET = TRUE

;ESPERAR CONFIRMACIÓN PALET CARGADO EN GARRAS
WHILE NOT PALET_COGIDO
    WAIT SEC 0.01
ENDWHILE
ENTERADO_PALET_CARGADO = TRUE

;CÁLCULO DE LA COTA DE SALIDA DEL ALIMENTADOR DE PALETS TRAS CARGAR
P ALIM SALIDA = $POS_ACT
P ALIM SALIDA.Z = P ALIM SALIDA.Z + ALT_SEG_BAJAR_LENTA

;COTA MÍNIMA PARA SALIR DEL ALIMENTADOR HACIA LA MESA DE PALETIZADO
Z_MIN_SAL ALIM PAL = P PAL APROX.Z + BASE_DATA[2].Z - BASE_DATA[1].Z

IF P ALIM SALIDA.Z < Z_MIN_SAL ALIM PAL THEN
    P ALIM SALIDA.Z = Z_MIN_SAL ALIM PAL
ENDIF

;SUBIR AL PUNTO DE SALIDA DEL ALIMENTADOR DE PALETS
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 100
TRIGGER WHEN DISTANCE = 0 DELAY = 50 DO PUNTO_CARGAR_PALET = FALSE
TRIGGER WHEN DISTANCE = 0 DELAY = 300 DO CERRAR_GARRA_TOMAR_PALET=TRUE
LIN P ALIM SALIDA C_DIS

;=====

```

```

;PALET CARGADO -> IR A MESA PALETIZADO A DESCARGAR
;=====
;CAMBIO A BASE PUNTOS MESA PALETIZADO
CONTINUE
$BASE = BASE_DATA[2]

;-----
;CÁLCULO DE PUNTOS DEPOSITADO PALET EN MESA PALETIZADO
;-----
;PUNTO DE ESPERA ANTES DE ENTRAR EN ÁREA DE MESA DE PALETIZADO
P_PAL_ESPERA = P_PAL_APROX
P_PAL_ESPERA.X = P_PAL_APROX.X - DIST_PAL_ESPERA
P_PAL_ESPERA.Z = P_ALIM_SALIDA.Z + BASE_DATA[1].Z - BASE_DATA[2].Z

;PUNTO DE SALIDA TRAS DEPOSITAR PALET EN MESA PALETIZADO
P_PAL_SALIDA = P_PAL_FINAL
P_PAL_SALIDA.Z = P_PAL_FINAL.Z + ALT_SEG_SALIDA_PAL

;=====
;TRAYECTORIA PUNTOS DEPOSITADO PALET EN MESA PALETIZADO
;=====
;-----
;IR A PUNTO DE ESPERA ENTRADA A MESA PALETIZADO (PALET EN GARRAS)
;-----
BAS(#VEL_PTP, 60)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO ROBOT_ALIMENTADOR = FALSE
PTP P_PAL_ESPERA C_PTP C_DIS

;ESPERAR PERMISO PARA ENTRAR CON PALET EN MESA PALETIZADO
CONTINUE
WHILE NOT PERMISO_ENTRAR_PALET
    WAIT SEC 0.01
ENDWHILE

;-----
;IR A PUNTO DE APROXIMACIÓN SOBRE MESA PALETIZADO
;-----
BAS(#VEL_PTP, 60)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
TRIGGER WHEN DISTANCE=0 DELAY=50 DO ROBOT_MESA = TRUE
PTP P_PAL_APROX C_PTP C_DIS

;-----
;IR A PUNTO PREDEPOSITADO PALET
;-----
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 50
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO CERRAR_GARRA_TOMAR_PALET = FALSE
TRIGGER WHEN DISTANCE=1 DELAY=0 DO PALET_PRE_DEPOSITADO = TRUE
LIN P_PAL_PREP C_DIS

;ESPERA PERMISO DEPOSITAR PALET SOBRE MESA PALETIZADO
CONTINUE
WHILE NOT PERMISO_DEJAR_PALET
    WAIT SEC 0.01
ENDWHILE

;-----
;IR A PUNTO FINAL DESPOSITADO PALET SOBRE MESA PALETIZADO
;-----
BAS(#VEL_CP, 0.1)
BAS(#ACC_CP, 100)
LIN P_PAL_FINAL

```

```
;ABRIR GARRA PARA DEPOSITAR PALET Y ESPERAR CONFIRMACIÓN
CERRAR_GARRA = FALSE
ABRIR_GARRA = TRUE
```

```
WHILE NOT GARRA_ABIERTA
    WAIT SEC 0.01
ENDWHILE
PALET_DEPOSITADO = TRUE
ENTERADO_PALET_CARGADO = FALSE
```

```
;ESPERAR CONFIRMACIÓN PALET DEPOSITADO
WHILE NOT CONF_PALET_DEPOSITADO
    WAIT SEC 0.01
ENDWHILE
PALET_DEPOSITADO = FALSE
```

```
;-----
;IR A PUNTO DE SALIDA TRAS DEPOSITAR PALET EN MESA PALETIZADO
;-----
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 100
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO ABRIR_GARRA = FALSE
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO CERRAR_GARRA = TRUE
TRIGGER WHEN DISTANCE = 1 DELAY= -10 DO ROBOT_MESA = FALSE
LIN P_PAL_SALIDA C_DIS
```

```
;ESPERAR CONFIRMACIÓN GARRA PALET CERRADA ANTES DE CONTINUAR
CONTINUE
WHILE NOT GARRA_CERRADA
    WAIT SEC 0.01
ENDWHILE
```

```
ENDIF
```

```
END
```



```

DEF coger1_paquete ( )
CONTINUE
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVACIÓN DE INTERRUPCIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;ACUSE AL PLC DE EJECUCIÓN DE ORDEN CARGAR PAQUETE DE LÍNEA 1
CONTINUE
PROG_LINEA1 = TRUE

;SI LA PINZA ESTÁ CARGADA -> IR A DEPOSITAR SOBRE EL PALET
CONTINUE
IF PAQUETE_COGIDO THEN
    dejar_paquete()

;SI LA PINZA ESTÁ DESCARGADA -> IR A CARGAR A LÍNEA 1
ELSE
    ;-----
    ;CAMBIO BASE REFERENCIA PUNTOS LÍNEA 1
    ;-----
    CONTINUE
    $TOOL = TOOL_DATA[1]
    CONTINUE
    $BASE = BASE_DATA[3]

    CONTINUE
    IF NOT VOLVER_A_CARGAR THEN
        INI_CALC_L1 = TRUE
        calc_ptos_lineal()
        posic_pinza_cargar()
    ENDIF

    ;-----
    ;IR A PUNTO DE ENTRADA DE CARGA EN LÍNEA 1
    ;-----
    CONTINUE
    IF IR_P_COJ1_ENTRAD THEN
        CONTINUE
        BAS(#VEL_PTP, 100)
        BAS(#ACC_PTP, 100)
        $APO.CPTP = 100
        $APO.CDIS = 500
    
```

```

    PTP P_COJ1_ENTRADA C_PTP C_DIS
ENDIF

;-----
;IR A PUNTO DE ESPERA PARA CARGAR
;-----
;ESPERA EL POSICIONAMIENTO CORRECTO DE LOS MECANISMOS DE LA PINZA
;(Antes de dirigirse al punto de espera)
CONTINUE
WHILE (NOT GARRA_CERRADA) OR (NOT PINZA_ABIERTA)
    WAIT SEC 0.01
ENDWHILE

CONTINUE
BAS(#VEL_PTP, 100)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO VOLVER_A_CARGAR = FALSE
PTP P_COJ1_ESPERA C_PTP C_DIS

;ESPERA PERMISO DE LÍNEA 1 PARA CARGAR PAQUETE
CONTINUE
WHILE NOT PERMISO_CARGAR_PAQUETE
    WAIT SEC 0.01
ENDWHILE

;-----
;DESCENSO AL PUNTO FINAL PARA CARGAR (PUNTO DE PARADA)
;-----
CONTINUE
BAS(#VEL_CP, 2)
BAS(#ACC_CP, 100)
TRIGGER WHEN DISTANCE=0 DELAY=50 DO ROBOT_LINEA1 = TRUE
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO PUNTO_CARGAR_PAQUETE = TRUE
LIN P_COJ1_FINAL

;CERRAR PINZA Y ESPERA CONFIRMACIÓN DE PAQUETE CARGADO
ABRIR_PINZA = FALSE
CERRAR_PINZA = TRUE

;ESPERA CONFIRMACIÓN PAQUETE COGIDO
WHILE NOT PAQUETE_COGIDO ;Espera confirmación de paquete cargado
    WAIT SEC 0.01
ENDWHILE
ENTERADO_PAQUETE_CARGADO = TRUE ;Acuse al PLC confirmación recibida

;-----
;SUBIR A PUNTO DE SALIDA UNA VEZ CARGADA LA PINZA
;-----
CONTINUE
BAS(#VEL_CP, 2)
BAS(#ACC_CP, 100)
$APO.CDIS = 400
TRIGGER WHEN DISTANCE=0 DELAY=50 DO PUNTO_CARGAR_PAQUETE = FALSE
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO ROBOT_LINEA1 = FALSE
LIN P_COJ1_SALIDA C_DIS

;ESPERA PERMISO PARA DEPOSITAR PAQUETE EN EL PALET
CONTINUE
WHILE NOT PERMISO_DEJAR_PAQUETE
    WAIT SEC 0.01
ENDWHILE

;TAREA IR A DEPOSITAR PAQUETE SOBRE MESA PALETIZADO
PAQUETE_L1 = TRUE
PAQUETE_L2 = FALSE
dejar_paquete()

```

ENDIF

END

```

DEF coger2_paquete ( )
CONTINUE
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVACIÓN DE INTERRUPCIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;ACUSE AL PLC DE EJECUCIÓN DE ORDEN CARGAR PAQUETE DE LÍNEA 2
CONTINUE
PROG_LINEA2 = TRUE

;SI LA PINZA ESTÁ CARGADA -> IR A DEPOSITAR SOBRE EL PALET
CONTINUE
IF PAQUETE_COGIDO THEN
    dejar_paquete()

;SI LA PINZA ESTÁ DESCARGADA -> IR A CARGAR A LÍNEA 2
ELSE
    ;-----
    ;CAMBIO BASE REFERENCIA PUNTOS LÍNEA 2
    ;-----
    CONTINUE
    $TOOL = TOOL_DATA[1]
    CONTINUE
    $BASE = BASE_DATA[4]

    CONTINUE
    IF NOT VOLVER_A_CARGAR THEN
        INI_CALC_L2 = TRUE
        calc_ptos_linea2()
        posic_pinza_cargar()
    ENDIF

    ;-----
    ;IR A PUNTO DE ENTRADA DE CARGA EN LÍNEA 2
    ;-----
    CONTINUE
    IF IR_P_COJ2_ENTRAD THEN
        CONTINUE
        BAS(#VEL_PTP, 100)
        BAS(#ACC_PTP, 100)
        $APO.CPTP = 100
        $APO.CDIS = 500
    
```

```

PTP P_COJ2_ENTRADA C_PTP C_DIS
ENDIF

;-----
;IR A PUNTO DE ESPERA PARA CARGAR
;-----
;ESPERA EL POSICIONAMIENTO CORRECTO DE LOS MECANISMOS DE LA PINZA
;(Antes de dirigirse al punto de espera)
CONTINUE
WHILE (NOT GARRA_CERRADA) OR (NOT PINZA_ABIERTA)
    WAIT SEC 0.01
ENDWHILE

CONTINUE
BAS(#VEL_PTP, 100)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 200
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO VOLVER_A_CARGAR = FALSE
PTP P_COJ2_ESPERA C_PTP C_DIS

;ESPERA PERMISO DE LÍNEA 2 PARA CARGAR PAQUETE
CONTINUE
WHILE NOT PERMISO_CARGAR_PAQUETE
    WAIT SEC 0.01
ENDWHILE

;-----
;DESCENSO AL PUNTO FINAL PARA CARGAR
;-----
CONTINUE
BAS(#VEL_CP, 2)
BAS(#ACC_CP, 100)
TRIGGER WHEN DISTANCE=0 DELAY=50 DO ROBOT_LINEA2 = TRUE
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO PUNTO_CARGAR_PAQUETE = TRUE
LIN P_COJ2_FINAL

;CERRAR PINZA Y ESPERA CONFIRMACIÓN DE PAQUETE CARGADO
ABRIR_PINZA = FALSE
CERRAR_PINZA = TRUE

;ESPERA CONFIRMACIÓN PAQUETE CARGADO
WHILE NOT PAQUETE_COGIDO
    WAIT SEC 0.01
ENDWHILE
ENTERADO_PAQUETE_CARGADO = TRUE ;Acuse al PLC confirmación recibida

;-----
;SUBIR A PUNTO DE SALIDA UNA VEZ CARGADA LA PINZA
;-----
CONTINUE
BAS(#VEL_CP, 2)
BAS(#ACC_CP, 100)
$APO.CDIS = 400
TRIGGER WHEN DISTANCE=0 DELAY=50 DO PUNTO_CARGAR_PAQUETE = FALSE
TRIGGER WHEN DISTANCE=1 DELAY=0 DO ROBOT_LINEA2 = FALSE
LIN P_COJ2_SALIDA C_DIS

CONTINUE
WHILE NOT PERMISO_DEJAR_PAQUETE
    WAIT SEC 0.01
ENDWHILE

;IR A DEPOSITAR PAQUETE SOBRE MESA PALETIZADO
PAQUETE_L2 = TRUE
PAQUETE_L1 = FALSE
dejar_paquete()

ENDIF

```

END

```

DEF inicio ( )

;ROBOT FUERA DE INTERFERENCIA CON OTROS MECANISMOS
ROBOT_ALIMENTADOR = FALSE
ROBOT_LINEA1 = FALSE
ROBOT_LINEA2 = FALSE
ROBOT_MESA = FALSE

;RESET ORDEN ABORTAR TAREA ACTIVA
TAREA_ABORTADA = TRUE

;RESET INDICADORES TAREA ACTIVA
PROG_EPALET = FALSE
PROG_DPALET = FALSE
PROG_LINEA1 = FALSE
PROG_LINEA2 = FALSE
PROG_HOME = FALSE

;RESET VARIABLES DE CONTROL INTERNO
VOLVER_A_CARGAR = FALSE

;ORDEN ABRIR PINZA PAQUETES SI NO ESTÁ CARGADA
IF NOT PAQUETE_COGIDO THEN
    CERRAR_PINZA = FALSE
    ABRIR_PINZA = TRUE
ENDIF

;ORDEN RECOGER GARRAS PALET SI NO ESTÁN CARGADAS
IF NOT PALET_COGIDO THEN
    EXTENDER_PALPADOR = FALSE
    RECOGER_PALPADOR = TRUE
    ABRIR_GARRA = FALSE
    CERRAR_GARRA = TRUE
ENDIF

;IR A CARGAR EN LÍNEA 1 Ó 2 SIN PASAR POR PUNTO DE ENTRADA
IR_P_COJ1_ENTRAD = FALSE
IR_P_COJ2_ENTRAD = FALSE

END

```

```

DEF dejar_paquete ( )
CONTINUE
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVACIÓN DE INTERRUPCIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;-----
;CAMBIO BASE REFERENCIA PUNTOS MESA PALETIZADO
;-----
CONTINUE
$BASE = BASE_DATA[2]
CONTINUE
$TOOL = TOOL_DATA[1]

;-----
;IR A PUNTO DE ENTRADA HACIA MESA DE PALETIZADO
;-----
CONTINUE
BAS(#VEL_PTP, 100)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
PTP P_DEJ_ENTRAD C_PTP

;*** Prueba de punto entrada a aproximación en trayectoria lineal *****
;CONTINUE
;BAS(#VEL_PTP, 100)
;BAS(#ACC_PTP, 100)
;$APO.CPTP = 100
;$APO.CDIS = 500
;PTP P_DEJ_ENTRAD C_PTP C_DIS

;-----
;IR A PUNTO DE APROXIMACIÓN A DESTINO DEPOSITADO PAQUETE
;-----
CONTINUE
BAS(#VEL_PTP, 100)
BAS(#ACC_PTP, 100)
$APO.CPTP = 100
$APO.CDIS = 500
PTP P_DEJ_APROX C_PTP C_DIS

;*** Prueba de punto entrada a aproximación en trayectoria lineal *****

```



```

;CONTINUE
;BAS(#VEL_CP, 2)
;BAS(#ACC_CP, 100)
;$APO.CDIS = 500
;LIN P_DEJ_APROX C_DIS

;-----
;IR A PUNTO PREVIO SOBRE DESTINO DEPOSITADO PAQUETE
;-----
CONTINUE
BAS(#VEL_CP, 2)
BAS(#ACC_CP, 100)
$APO.CDIS = 100
TRIGGER WHEN DISTANCE=0 DELAY=50 DO ROBOT_MESA = TRUE
LIN P_DEJ_PREVIO C_DIS

;-----
;IR A PUNTO FINAL DESTINO DEPOSITADO PAQUETE (FIN TRAYECTORIA)
;-----
CONTINUE
BAS(#VEL_CP, 1.5)
BAS(#ACC_CP, 100)
TRIGGER WHEN DISTANCE=1 DELAY=0 DO PUNTO_DESCAR_PAQUETE = TRUE
LIN P_DEJ_FINAL

;ABRIR PINZA PARA DEPOSITAR EL PAQUETE SOBRE EL PALET
CERRAR_PINZA = FALSE
ABRIR_PINZA = TRUE

;ESPERA CONFIRMACIÓN PINZA ABIERTA
WHILE NOT PINZA_ABIERTA
    WAIT SEC 0.01
ENDWHILE

;INFORME AL PLC DE PAQUETE DEPOSITADO
PAQUETE_DEPOSITADO = TRUE

;RETIRA EL ACUSE DE ENTERADO DE PAQUETE CARGADO
ENTERADO_PAQUETE_CARGADO = FALSE

;ESPERA CONFIRMACIÓN DE PLC DE PAQUETE DEPOSITADO (PINZAS ABIERTAS)
WHILE NOT CONF_PAQUETE_DEPOSITADO
    WAIT SEC 0.01
ENDWHILE

;ACUSE AL PLC CONFIRMACIÓN RECIBIDA
PAQUETE_DEPOSITADO = FALSE

;*****
;CÁLCULO PUNTOS TRAYECTORIA CARGA-DESCARGA PAQUETE SIGUIENTE
;-----
WAIT SEC 0.2 ;Retardo para asegurar los nuevos datos recibidos del PLC

WHILE NOT (PAQUETE_L1 OR PAQUETE_L2)
    WAIT SEC 0.01
ENDWHILE

IF PAQUETE_L1 THEN
    calc_ptos_lineal()
ELSE
    calc_ptos_linea2()
ENDIF

PAQUETE_L1 = FALSE
PAQUETE_L2 = FALSE

;*****
;-----

```

```
;IR A PUNTO DE SALIDA TRAS DEPOSITAR PAQUETE SOBRE EL PALET
```

```
;
```

```
CONTINUE
```

```
BAS(#VEL_CP, 2)
```

```
BAS(#ACC_CP, 100)
```

```
$APO.CDIS = 400
```

```
TRIGGER WHEN DISTANCE=0 DELAY=50 DO PUNTO_DESCAR_PAQUETE = FALSE
```

```
TRIGGER WHEN DISTANCE=1 DELAY=-50 DO ROBOT_MESA = FALSE
```

```
LIN P_DEJ_SALIDA C_DIS
```

```
;
```

```
;IR A PUNTO DE ENTRADA MESA PALETIZADO SI HAY NUEVA ORDEN DE CARGA
```

```
;
```

```
CONTINUE
```

```
IF (CARGAR_LINEA1 OR CARGAR_LINEA2) THEN
```

```
CONTINUE
```

```
BAS(#VEL_PTP, 100)
```

```
BAS(#ACC_PTP, 100)
```

```
$APO.CPTP = 100
```

```
$APO.CDIS = 500
```

```
TRIGGER WHEN DISTANCE=0 DELAY=50 DO VOLVER_A_CARGAR = TRUE
```

```
PTP P_DEJ_VOLVER C_PTP C_DIS
```

```
ENDIF
```

```
END
```

```
DEF palet_detectado ( )

;DESACTIVAR INTERRUPCIÓN PALPADOR PALET (UNA VEZ ATENDIDA)
INTERRUPT OFF 4

;1. DETENER EL MOVIMIENTO DEL ROBOT
BRAKE

;2. RECOGER EL BRAZO DEL PALPADOR DE PALET (TRAS PALPAR)
CONTINUE
EXTENDER_PALPADOR = FALSE
CONTINUE
RECOGER_PALPADOR = TRUE

;3. GUARDAR LA POSICIÓN ACTUAL DE PALPACIÓN DEL PALET
; (Para uso en siguiente búsqueda de palet)
CONTINUE
P_POS_PALET = $POS_ACT

;4. DESCENDER UN POCO MÁS PARA CARGAR EL PALET DETECTADO
;LIN_REL {Z - 20}

;5. INFORMAR QUE EL ROBOT ESTÁ POSICIONADO EN EL PUNTO DE CARGA DE PALET
PUNTO_CARGAR_PALET = TRUE

RESUME
END
```

```
DEF posic_garra_cargar ( )

;=====
;POSICIONAR LOS MECANISMOS DE LA GARRA PARA CARGAR PALET
;=====

;ABRIR GARRA PARA COGER PALET
CONTINUE
CERRAR_GARRA_TOMAR_PALET=FALSE
CONTINUE
CERRAR_GARRA = FALSE
CONTINUE
ABRIR_GARRA = TRUE

;EXTENDER PALPADOR DE PALETS
CONTINUE
RECOGER_PALPADOR = FALSE
CONTINUE
EXTENDER_PALPADOR = TRUE

END
```

```
DEF posic_pinza_cargar ( )

;=====
;POSICIONAR LOS MECANISMOS DE LA PINZA PARA CARGAR PAQUETE
;=====

;RECOGER PALPADOR DE PALETS
CONTINUE
EXTENDER_PALPADOR = FALSE
CONTINUE
RECOGER_PALPADOR = TRUE

;RECOGER GARRAS PALET
CONTINUE
CERRAR_GARRA_TOMAR_PALET = FALSE
CONTINUE
ABRIR_GARRA = FALSE
CONTINUE
CERRAR_GARRA = TRUE

;ABRIR_PINZA PAQUETES
CONTINUE
CERRAR_PINZA = FALSE
CONTINUE
ABRIR_PINZA = TRUE

END
```

```

DEF principal ( )
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS (#INITMOV,0 )

USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV()
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea()
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----

;=====
;SECCIÓN DE INICIALIZACIÓN (PROGRAMA PRINCIPAL)
;=====

$BASE = $NULLFRAME      ;Sistema XYZ inicial
$TOOL = TOOL_DATA[1]    ;Base de la Herramienta

PTP $POS_ACT ;Primer movimiento del Robot obligatorio en PTP

;DESDE POSICIÓN ACTUAL TOMAR COTA SEGURA ANTES DE IR A POSICIÓN "HOME"
P_SEG_HOME = $POS_ACT
P_SEG_HOME.Z = COTA_Z_SEGURA ;Cota segura elevación antes de llevar Robot a HOME

;-----
;IR A PUNTO EN ASCENSO A COTA SEGURA DESDE POSICIÓN ACTUAL
;-----
BAS(#VEL_CP, 1)
BAS(#ACC_CP, 100)
$APO.CDIS = 200
;Inicializar las condiciones de partida del Robot
TRIGGER WHEN DISTANCE=1 DELAY=0 DO inicio() PRIO=-1
LIN P_SEG_HOME C_DIS

;LLEVAR A POSICIÓN "HOME" (UNA VEZ EN COTA SEGURA)
PTP HOME Vel= 90 % DEFAULT
    $BWDSTART = FALSE
    PDAT_ACT=PDEFAULT
    FDAT_ACT=FHOME
    BAS (#PTP_PARAMS,90 )
    $H_POS=XHOME
    PTP XHOME

;-----
;DECLARACIÓN DE INTERRUPCIÓN "ABORTAR TAREA EN CURSO"
;-----
;Hay que activar la interrupción en cada uno de los subprogramas
;$CYCFLAG[2] = ABORTAR_TAREA
;INTERRUPT DECL 2 WHEN $CYCFLAG[2] DO abortar_tarea()

;=====
;SECCIÓN CÍCLICA (PROGRAMA PRINCIPAL)
;=====
LOOP

```

```

;ACTIVACIÓN DE INTERRUPCIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;SI ORDEN CARGAR PAQUETE EN LÍNEA 1 Y DEPOSITAR EN PALET
CONTINUE
IF CARGAR_LINEA1 THEN
    coger1_paquete()
ENDIF

;SI ORDEN CARGAR PAQUETE EN LÍNEA 2 Y DEPOSITAR EN PALET
CONTINUE
IF CARGAR_LINEA2 THEN
    coger2_paquete()
ENDIF

;SI ORDEN CARGAR PALET TIPO 1 (EUROPALET) DEL ALIMENTADOR
CONTINUE
IF TIPO_EUROPALET THEN
    EUROPALET = TRUE
    coger_palet()
ENDIF

;SI ORDEN CARGAR PALET TIPO 2 (PALETA GRANDE) DEL ALIMENTADOR
CONTINUE
IF TIPO_DOBLEPALET THEN
    EUROPALET = FALSE
    coger_palet()
ENDIF

;SI ORDEN DE LLEVAR EL ROBOT A POSICIÓN "HOME"
CONTINUE
IF ROBOT_HOME THEN
    robot_home()
ENDIF

ENDLOOP
END

```

```

DEF robot_home ( )
IF $MODE_OP == #T1 THEN
INI
BASISTECH INI
    GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
    INTERRUPT ON 3
    BAS ( #INITMOV,0 )
USER INI
    ;Make your modifications here
    BASE_ALIMENT = BASE_DATA[1]
    BASE_PALET = BASE_DATA[2]
    BASE_LINEA1 = BASE_DATA[3]
    BASE_LINEA2 = BASE_DATA[4]
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA DETENER EL MOVIMIENTO DEL ROBOT
    $CYCFLAG[1] = PER_MOV_ROBOT
    GLOBAL INTERRUPT DECL 1 WHEN $CYCFLAG[1]==FALSE DO STOP_MOV( )
    INTERRUPT ON 1
    WAIT SEC 0.012
    ;-----
    ;DECLARACIÓN DE INTERRUPCIÓN PARA ABORTAR TAREA EN CURSO
    $CYCFLAG[2] = ABORTAR_TAREA
    INTERRUPT DECL 2 WHEN $CYCFLAG[2]==TRUE DO abortar_tarea( )
    INTERRUPT ON 2
    WAIT SEC 0.012
    ;-----
ENDIF

;ACTIVACIÓN DE INTERRUPCIÓN ABORTAR TAREA ACTIVA
;CONTINUE
;INTERRUPT ON 2

;INFORME EJECUCIÓN TAREA IR A POSICIÓN HOME
PROG_HOME = TRUE

$BASE = $NULLFRAME
$TOOL = TOOL_DATA[1]

;PRIMER MOVIMIENTO PTP (OBLIGADO)
PTP $POS_ACT

;CÁLCULO PUNTO ASCENSO A COTA SEGURA
P_SEG_HOME = $POS_ACT
P_SEG_HOME.Z = COTA_Z_SEGURA

;ASCENSO VERTICAL A COTA SEGURA
BAS( #VEL_CP, 1 )
$APO.CDIS = 30
LIN P_SEG_HOME C_DIS

;LLEVAR ROBOT A POSICIÓN "HOME"
PTP HOME Vel= 90 % DEFAULT
    $BWDSTART = FALSE
    PDAT_ACT=PDEFAULT
    FDAT_ACT=FHOME
    BAS ( #PTP_PARAMS,90 )
    $H_POS=XHOME
    PTP XHOME

END

```



```
DEF STOP_MOV ( )

INTERRUPT OFF 1
ROBOT_DETENIDO = TRUE

IF ( $CYCFLAG[1]==FALSE ) THEN
;DETENER EL MOVIMIENTO DEL ROBOT
BAS(#ACC_PTP, 50)
BAS(#ACC_CP, 50)
BRAKE

REPEAT
POWER=SYNC( )
HALT
UNTIL ( $CYCFLAG[1]==TRUE ) AND ( $STOPMESS==FALSE ) AND ( $POWER_FAIL==FALSE )

ENDIF

INTERRUPT ON 1
ROBOT_DETENIDO = FALSE

END
```